Microscale Simulation of the Mechanical and Electromagnetic Behavior of Textiles

by

Alejandro Francisco Queiruga

A thesis submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Mechanical Engineering
and the Designated Emphasis
in
Computational Data Science and Engineering

in the

Graduate Division
of the
University of California, Berkeley

Committee in charge:

Professor Tarek I. Zohdi, Chair
Professor Per-Olof Persson
Professor David J. Steigmann

Spring 2015

Microscale Simulation of the Mechanical and Electromagnetic Behavior of Textiles

Abstract

Microscale Simulation of the Mechanical and Electromagnetic Behavior of Textiles

by

Alejandro Francisco Queiruga
Doctor of Philosophy in Engineering - Mechanical Engineering

University of California, Berkeley

Professor Tarek I. Zohdi, Chair

A computational framework for assisting in the development of novel textiles is presented. Electronic textiles are key in the rapidly growing field of wearable electronics for both consumer and military uses. Fabric actuators can be made with electrically functionalized fabrics that can be manipulated by externally applied electromagnetic fields when electric current is run through the yarns of the fabric. There are two main challenges to the modeling of electronic textiles: the discretization of the textile microstructure and the interaction between electromagnetic and mechanical fields.

The fully coupled mechanical, thermal, and electromagnetic behavior of a textile can be simulated in the context of quasistatic material property prediction and dynamic analysis of high speed impacts. Director-based beam formulations are used to discretize the fabric at the level of individual fibrils. Instead of solving Maxwell's equations in full detail, a quasistatic approximation is used to solve the electric potential in the presence of a moving material medium. While this formulation alleviates the spatial and temporal discretization restrictions, the coupled problem is a Differential Algebraic Equation requiring special treatment. Diagonally Implicit Runge-Kutta methods using a monolithic Newton's method solver are used to integrate the resulting nonlinear coupled systems in time. The finite element model is implemented using the open source package FEniCS. Contact integrals were added into the FEniCS framework so that multiphysics contact laws can be incorporated in the same framework, leveraging the code generation and automatic differentiation capabilities of FEniCS to produce the tangents needed by the implicit solution method.

The nonlinear deformation of a current-carrying elastic string is solved analytically. The computational model for a single fibril is validated using by comparison the static problem and verifying the convergence orders for higher-order finite element basis functions. The time stepping method for the fully coupled differential algebraic equation is verified using the convergence orders of the higher-order Runge-Kutta methods. The computational model is used to construct and determine the mechanical, thermal, and electrical properties of representative volume elements of textiles using dynamic relaxation to solve the decoupled fields in a static context. The dynamic deformation of a small electronic textile under various orientations of magnetic fields is solved. An electromagnetically-enhanced textile armor system impacted by a projectile is simulated.

To my parents, Elena and Francisco.

Contents

List of Figures

List of Tables

List of Algorithms

Acknowledgments

The author would like to thank Professor Zohdi for his years of advice. In addition to his guidance through courses and research, his guidance while serving as his teaching assistant has been valuable. The author also thanks his committee members Professor Persson and Professor Steigmann, whose instruction and advice was invaluable to the author's understanding of many difficult topics. Much of the electromagnetic theory for materials that this work is built on is due to the course Electrodynamics of Continuous Media at UC Berkeley taught by Professor Steigmann. The author would also like to thank Doctor Namburu at the Army Research Lab, whose input helped shape the direction of this research.

This work was supported by funding from the Army High Performance Computing Research Center and Army Research Lab.

Chapter 1

Introduction

1.1 Motivation

Textiles are one of the most common materials used and have widespread applications. They are characterized by their multiscale structure, wherein a single sheet of fabric is composed of a woven or knitted network of yarns, which are in turn composed of spun fibrils. Fabrics with novel mechanical, thermal, or electrical properties can be created by weaving different types of yarns made from different types of fibrils in various patterns. This work is directly interested in textiles that are functionalized using conductive yarns created by spinning metal fibrils together with high strength fibrils. The resulting textile can be manipulated by externally applied electric and magnetic fields when an electric current is run through the yarns of the fabric. Electronic textiles have applications as electronic substrates, active armors, energy harvesting devices, and actuators. Designing new textile materials with exotic properties requires models with predictive capabilities for the multiphysics behavior.

This work is driven by the model problem of designing novel electromagnetically-enhanced armors employing electronic-textiles. High strength textiles are a fundamental component of armors in multiple applications, where they may be supported by other components such as metal and ceramic plates. Many nontraditional strategies for improving the effectiveness of armors are possible. For example, reactive armors—where a counter-explosive is detonated against an incoming projectile—have been pivotal in mechanized warfare. Even more strategies are possible, such as embedding ballistic textiles in shear thickening fluids to increase energy dissipation [78]. Prospective applications of electromagnetic effects in armor systems have been explored. For example, non-explosive electromagnetic launchers replacing explosive reactive armor have been demonstrated [23, 108, 122, 127], as well as passive systems designed to negate the effectiveness of armor-piercing shaped chargers [59, 119]. One such configuration for making use of electronic textiles in a ballistic system is diagrammed in Figure 1.1.2. The high-speed and destructive nature of armor systems allows for the exploration of effects of electromagnetic systems not normally observed from the perspectives of the analyses of the theoretical formulations, applications of numerical methods, and engineering design.

In the following chapters, a computational model for the dynamic behavior of electromagnetically functionalized textiles is developed. The textile microstructure is discretized using

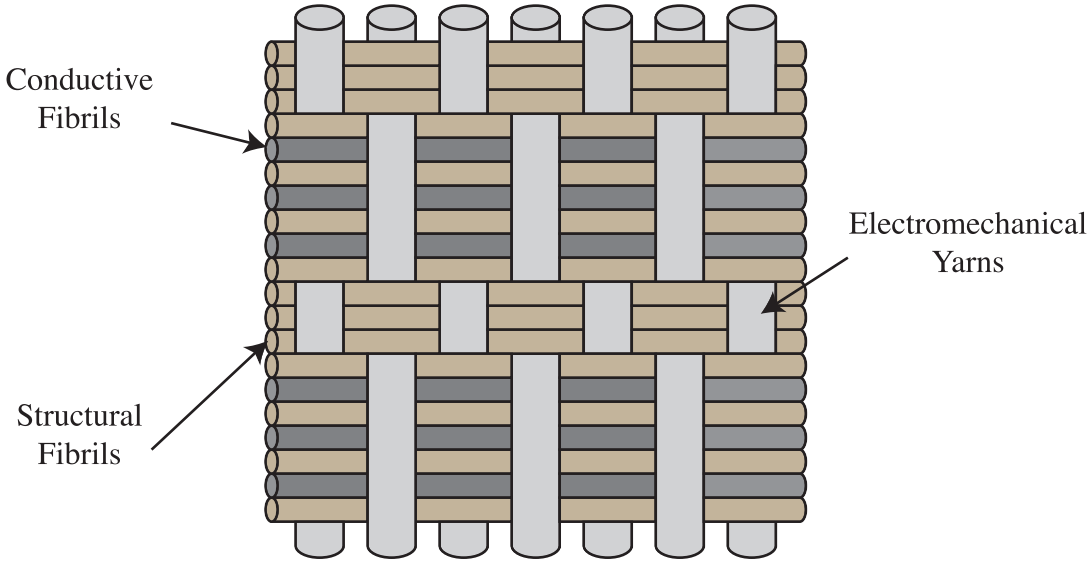


Figure 1.1.1: Example of a functionalized electronic textile with actuating yarn in the vertical direction and conductive and structural yarns in the horizontal direction. See [52] for a realized textile actuator of this design.

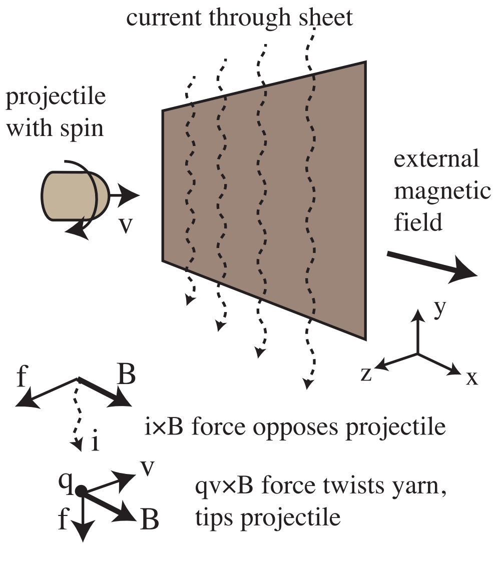


Figure 1.1.2: Projectile interacting with an electrified ballistic fabric as an example armor configuration.

a director-based beam model on individual fibrils with thermal and electrical fields incorporated. The electromagnetic problem is simplified to yield a differential algebraic equation (DAE) when coupled to the partial differential equations of continuum mechanics. Complex multiphysics problems are difficult to discretize and program. To assist in development, the package FEniCS, a Domain Specific Language (DSL), is used to create the finite element calculations for the complex fully coupled nonlinear equations. The capabilities of the package were extend to automatically generate code for contact integrals that are needed for the textile model. Contact integrals are formulated in a manner that can be easily represented by the existing DSL syntax for variational forms. The computational model is used to predict the mechanical and electrical properties of textile materials using representative volume elements (RVEs) and simulate the dynamic behavior of a textile-based ballistic armor.

A brief overview of existing electronic textile research and design is provided in Section 1.2. The methods that have been used to analyze textiles, focusing on ballistic applications, are reviewed in Section 1.3. Works in the simulation of electromagnetic structure interaction are reviewed in Section 1.4. The programming methodology of using a Domain Specific Language for code generation is described in Section 1.5. Finally, Section 1.6 outlines the rest of the chapters in this work.

1.2 Electronic Textiles

Textiles are becoming a regular substrate for electronic devices. Circuits can be placed onto fabrics by embroidering conductive yarns onto standard fabrics [84, 100], using conductive paints [20, 68], or weaving conductive elements into the textile [26, 46, 73, 114]. Traditional electrical devices have been attached to textile circuits using different methods, such stitching through the pad of a standard circuit board [83, 84] or flexible circuit boards [69], or bending a through-hole component into a sequin or soldering beads onto the ends of surface mount device [19]. Circuit layouts from standard printed circuit board CAD formats were automatically embroidered in [42]. Development kits for electronic textiles accessible to children are even available with Arduino-based platforms such as a the Lilypad [18, 21] or the FLORA [44].

The capability to embed complex electronic devices onto clothing has been demonstrated, such as EKG pads for medical sensing [82], embroidered antennae [88], and even standard USB data lines [120]. Electromechanical properties of yarns were studied experimentally in [54] to quantify the signal transfer performance of structural yarns for composites. Organic transistors have been placed between yarns in [76, 77] and [53] to create fabric circuits capable of executing simple logic. Woven capacitors were demonstrated in [116]. User-interface devices were embroidered into textiles in [47]. Textile based strain sensors have been illustrated in [30, 24]. Analysis of the electrical properties of knitted electrical textiles under mechanical strain was performed in [111, 124, 125]. Functionalized wearable textiles capable of sensing ambient gasses and optical environmental states were demonstrated in [81]. Textiles are also useful substrates for non-wearable devices. For examples, the flexible nature was used to create a large deployable beam forming array in [92]. A review of electronic textile applications can be found in [89].

Heterogenous textiles provide a way of combining yarns made from different materials

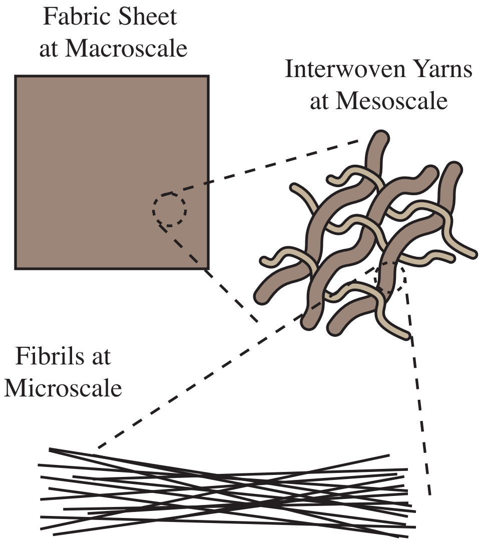


Figure 1.3.1: Multiscale structure of fabrics

into usable devices. Besides conductive elements, more exotic materials have been incorporated into fabrics to add functionality. In [52], a thermomechanical fibril was developed. To create an easy to use actuator, the new fibrils were woven together using cotton yarns for structural support and conductive yarns to heat up and activate the constriction of the thermomechanical fibrils. Displays were created using thermochromic inks in [10]. Piezoelectric fibrils have been made using electrospinning techniques [74]. Textiles incorporating such yarns have applications in force sensing [73] and energy harvesting [114]. In [123], functionalized yarns were made by using a conductive coating.

1.3 Analysis of Fabrics

Analytical relations for the behavior of ballistic fabrics and armors in impact have been formulated[12, 15, 27, 28, 93, 98, 99, 115]. Analytical formulae for the properties of yarns from their microscale structure in the context of wire ropes were developed in [31]. While such equations can be useful, it is difficult to analytically incorporate complicated effects, such as multi-physics effects as in the couple of shear-thickening fluids or electromagnetism. Examples of experimental studies of ballistic fabrics can be seen in [96, 97, 105]. The effects of friction within the textile microstructure have been studied [17, 71, 70]. However, experimental tests are costly and time consuming due to their inherently destructive nature. The reader is directed to [25, 109, 33] for more exhaustive reviews of studies as well as existing designs of ballistic armors.

Numerical simulations of ballistic fabrics are thus an appealing alternative, because various types and configurations of armor systems can be easily tested quickly and with little cost. However, difficulties arise in forming computational models of textiles due to the multiple length scales of structure present, diagrammed in Figure 1.3.1. Standard continuum mechanics based finite element analysis requires too fine a mesh to capture the complete

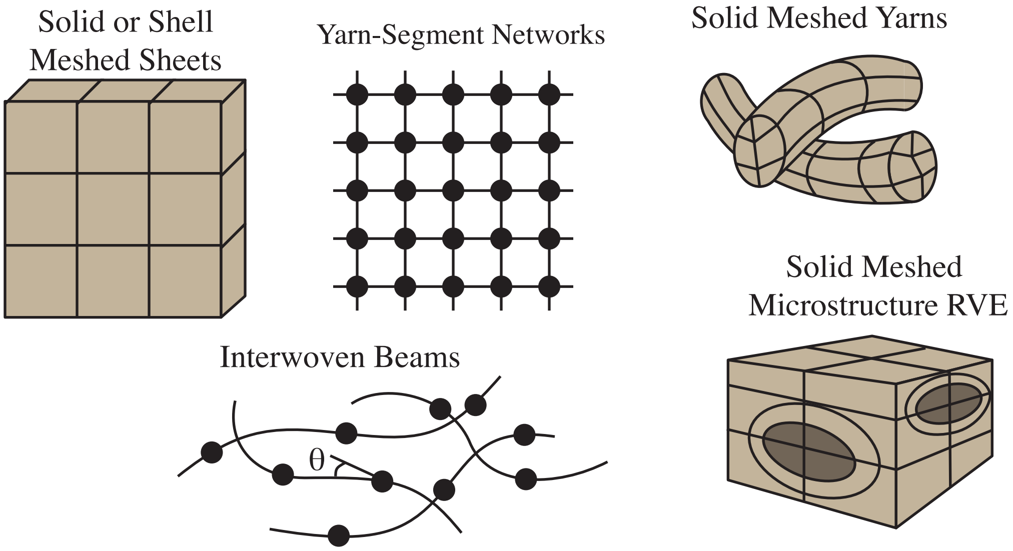


Figure 1.3.2: Fabric discretization types.

microstructure to be computationally efficient. The regular microstructure of woven fabrics yields anisotropic properties (non-woven fabrics with random microstructures, such as felt, can be isotropic.) The orthotropic nature of textiles is significant in determining the overall behavior. In ballistic applications, the transverse wave front has a distinctive diamond-like shape in a woven textile [105], resulting in the damage in the fabric spreading along the yarns.[109, 130] For ballistic textiles, a yarn may contain on the order of eighty individual fibrils. Many approaches have been taken to model the fabric microstructure. The discretization level in the microstructure is the primary distinguishing feature of a computational model of textiles. The discretization methods can be categorized as follows:

1. Macroscale continuum discretization using shell or thin solid finite elements with homogenized properties
2. Mesoscale discretization:
   (a) Network models of connected yarn segments
   (b) Beam models of yarns with effective responses
   (c) Solid meshed yarns with homogenized continuum models
3. Microscale discretization:
   (a) Solid meshing of microstructure for multiscale modeling
   (b) Beam modeling of the fibril microstructure

These categories are illustrated in Figure 1.3.2.

Multi-scale finite element simulations have been used to reduce the degrees of freedom, where a fully discretized unit cell of woven yarns is used to provide the response for plate and shell elements [91, 95]. Highly local effects such as the damage and slipping of yarns due to a projectile piercing the fabric or global properties such as the current along a single damaged yarn are difficult to represent with this approach.

Finite element simulations with the individual yarns discretized with solid meshes of the textiles have been performed extensively [37, 50]. In [13], image processing techniques were used to post process the strains in fabrics, and corresponding finite element models of woven and knitted textiles with meshed yarns were performed. Impact simulations using fully meshed interwoven yarns were performed in [50, 106]. For example, finite element models of textiles alone and embedded in composite matrix materials with different weave patterns under impact were performed in [67]. While the detail of such a discretization allows for many properties to be modeled, the inherent geometric structure of the fabric forces a large amount of degrees of freedom for even a small piece of fabric. A multiscale simulation of an impact meshing a large sheet as solid yarns near the impact site, coupled to interwoven shell elements, coupled to a membrane model for the far field effects in [95].

Discretizing the individual fibrils of a yarn is prohibitively expensive, thus some empirical or analytical relation is required as a material model for the yarn. Multiscale models with meshed fibril microstructures have been performed in [29]. An analytical solution of interlocking yarn beams is used as a constitutive model for a membrane finite element simulation in [91]. In [43], the fibril microstructure is meshed fully to produce a representative volume element of braided rope and used to produce effective properties for a beam model of the resulting yarn.

In order to preserve the global and local properties of the woven yarn structure, while still reducing the degrees of freedom required, lattice-network models are employed. In such methods, the woven yarns are modeled as nodes interconnected by “yarn segments”, which can be viewed as one-dimensional linear-elements, trusses, or springs (see Figure 1.3.2). A finite-element oriented approach for this technique is seen in [110]. An analytical network model for textiles under impact can be seen in [79]. Empirical damage laws and electromagnetic effects were incorporated into a network model in [132, 130, 131]. Advanced summation techniques for designed for electronic textiles and applied to atomistic lattice models was developed in [7]. Similar techniques have been applied to the analysis of cable networks, such as the methods in [32, 22].

In [41], the stresses within yarn were analyzed with a fibril-level simulation and it was concluded that a homogenized continuum yarn representation would be inadequate. Two sets of methods employ a fibril-level microscale discretization that are similar to that used in this work.

The multi-chain digital element method represents individual fibrils as pin-jointed rod elements[118, 129]. The chains do not model the bending stiffness of the material and are given an artificial lateral stiffness to alleviate the numerical issues associated with this lack of stiffness. (As will be discussed in Chapter 6, there are still numerical issues with the low bending stiffness of the fibrils even using beam models.) The method is capable of static and sliding frictional contacts using a rod-rod contact geometry[90]. The digital elements were coupled with a vacuum bag model to simulate composite manufacturing in [128]. The method was used in [117] to perform a ballistic impact simulation on a fully discretized textile sheet with up to 19 fibrils per yarn in a 33mm by 33mm sheet with approximately 44 yarns in each direction.

Fibrils are assembled into woven structures using a nonlinear beam description in [39, 40, 41] The same kinematic beam description is used in this work and will be described in Chapter 6. This method has been used the study of non-woven fabrics, [39], biological tissue engineering [75], and knot-forming [38]. In a similar vein to the goals of this work, the methodology has also been used to analyze the fatiguing of superconducting wires due to stresses arising from thermal and electrical loads [3, 112].

1.4 Electromagnetic Structure Interaction Modeling

Electromagnetism provides a similar scaling challenge as the textile microstructure when designing computational models. Even in the context of high speed ballistic impacts of very small microstructures, the electromagnetic fields have very disparate spatial and temporal time scales to the mechanical fields. Finite difference methods discretizing Maxwell's equations directly are extremely popular for electromagnetism [61]. The classic nodal finite element shape functions behave poorly for discretizing $\mathbf{E}$ and $\mathbf{B}$ and solving Maxwell's equations, but specially designed elements have been designed to enforce the curl-type and divergence-type nature of the fields from the onset [94]. However, methods solving Maxwell's equations directly require timesteps on the order of $10^{-14}s$ and similarly scaled spatial grids to discretize individual electromagnetic waves [61]. The finite element method is applied to solve the magnetic vector potential coupled to a saturable permeable material subject to deformations in [102, 11]. An discontinuous Galerkin enhanced immersed boundary finite element method for solving the electric potential with material discontinuities is presented in [16] that is designed for problems in structure-interaction. The $\mathbf{E}$ and $\mathbf{B}$ fields are discretized with the large deformation of a thin membrane in [4]. Finite element analysis of the electromagnetic fields has been used to model active electromagnetic launcher armors [23, 122, 127] and passive electromagnetic armors [126].

1.5 Domain Specific Languages

High performance scientific codes for simulating physical systems, such as solving partial differential equations, have two sides:

1. Performing a calculation on a small chunk data: the physical kernel
2. Distributing the kernel calculation and required data across a large amount of resources for parallel computation

For a finite element method program, the calculation of the local element matrix is the kernel that is the smallest unit of computational work to be distributed across processing nodes. Developing algorithms and working code on both sides of the program is a challenge. During the development of the computational model, this work focused on alleviating the difficulties in the first of these points.

Higher order numerical algorithms for both temporal discretizations, such as many-stage implicit Runge-Kutta methods, and spatial discretizations, such as high order finite element basis functions, are, in the author's opinion, underused owing to the great difficulties in their implementation. A major barrier to using the more complex numerical schemes is the generation of the tangent matrix: that is, the $\mathbf{K}$ in the problem $\mathbf{K}\mathbf{u} = \mathbf{f}$. Developing the form of the matrix is manageable for linear problems, though quickly becomes difficult for nonlinear fully-coupled multiphysics problems. These types of problems are typically described mathematically as either minimization problems on a Lagrangian or some other expression of a potential,

$$\min_{\mathbf{u}} \Pi\left(\mathbf{u}\right),$$

or as nonlinear systems of equations,

$$\mathbf{f}\left(\mathbf{u}\right) = 0.$$

Solving these problems statically or with an implicit time stepping method usually centers around linearizing the functions and employing Newton's method, or some variant thereof, and iterating over a series of linear systems,

$$\mathbf{f}\left(\mathbf{u}^I\right) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\mathbf{u}^I} \Delta\mathbf{u}^I \;\; = \;\; 0,$$

with $\mathbf{f} = \frac{\partial \Pi}{\partial \mathbf{u}}$ if the problem was originally expressed with a potential. Even neglecting the effort required to write the code itself, it can require many weeks of pencil-and-paper work manipulating the mathematical expressions to produce a linearized equation. Explicit time stepping schemes are thus very popular for time-dependent nonlinear problems, and can perform very well [8]. However, complex constraints, such as the treatment of electromagnetic fields in this work, adds implicit components cannot be side-stepped by an explicit scheme (discussed in Chapters 2 and 6.)

The theory behind many numerical methods, especially the finite element method, has been developed to the point where there is a robust “turn-the-crank” methodology for generating a working discretization. Computer programs can be written to automate steps in the development of a numerical code that can be employed in different ways to produce different workflows, illustrated in Figure 1.5.1. Domain Specific Languages (DSLs) provide a specialized environment for implementing numerical codes of a certain type. Common examples are Matlab, which is tailored for linear algebra, and R, which is tailored for statistics.

To achieve high performance, the DSL compiler needs to compile the calculations into machine-level code. The equations are specified in a high-level language, i.e. a DSL, that is parsed by the compiler and translated into an intermediate code in a low-level language, such as C, that is passed into the “real” compiler (such as gcc) to be turned into machine code. This yields a meta-programming style, in which one writes code to write code. One of the more popular combinations of tools is implementing the compiler in Python and have it write C/C++ code that is compiled and dynamically linked back into the Python code [9, 86]. The programming language Terra is specifically designed with the goal of seamlessly implementing these types of languages [35].

In addition to translating physical equations into usable computer code, it is also desirable to generate the expressions or code for the tangents to these equations. There are three methods for the computerized generation of tangents:

1. numerical differentiation via divided differences,
2. symbolic differentiation on mathematical expressions, and
3. automatic differentiation.

Numerical differentiation is the easiest to implement, where one simply uses finite difference stencils on the input to the code, but is the least stable and least efficient [49]. (It can be a useful method to employ for comparing against buggy handwritten tangent code.)

Traditional workflow with all steps done by hand:

$$f\left(u\right) \longrightarrow \underset{\text{Derivative}}{\frac{\partial f}{\partial u}} \longrightarrow \underset{\text{Discretization}}{f\left(u=\sum a\phi\right)} \longrightarrow \underset{\text{Low-level Code}}{\texttt{eval\_f\_\&\_df}\left(\texttt{a}\right)\{...\}}$$

Workflow employing automatic differentiation of the source code:

$$f\left(u\right) \longrightarrow \underset{\text{Discretization}}{f\left(u=\sum a\phi\right)} \longrightarrow \underset{\text{Low-level Code}}{\texttt{eval\_f}\left(\texttt{a}\right)\{...\}} \Longrightarrow \underset{\text{Autogenerated Code}}{\texttt{eval\_df}\left(\texttt{a}\right)\{...\}}$$

Workflow employing a DSL for code generation:

$$f\left(u\right) \longrightarrow \underset{\text{DSL Code}}{\texttt{f}\left(\texttt{u}\right)*\texttt{dx}} \Longrightarrow \underset{\text{Auto Discretization}}{u=\sum a\phi} \Longrightarrow \underset{\text{Auto Derivative}}{\frac{\partial f}{\partial a}} \Longrightarrow \underset{\text{Autogenerated Code}}{\texttt{eval\_f\_\&\_df}\left(\texttt{a}\right)\{...\}}$$

Figure 1.5.1: Comparison of workflows for creating a numerical code for some function $f\left(u\right)$ with some discretization illustrated by $\sum a\phi$. Solid arrows indicate steps taken by the human, and double arrows indicate steps taken by a computer program. The final low-level code is compiled into machine code in all cases illustrated.

Automatic differentiation is employed on a computer program directly to modify the original program's execution or create a new program that also calculates the derivative. The methods themselves are outside of the scope of this work; the reader is directed to [49] for a discussion on automatic differentiation. Many computer packages exist that employ symbolic differentiation to assist in analytical mathematics, such as Mathematica, but these are not tailored to creating high performance codes. (These packages can be used for code generation; Mathematica was employed many times for minor things in this work.)

OpenFOAM is a numerical package for Finite Volume Method, where the user describes the strong-form PDE in an application-specific syntax that is used to automatically create the corresponding FVM (or FEM) code [64]. Liszt is another such language where the program describes discretizations and operations to be applied a mesh that can be compiled into high performance CPU or GPU code [34, 36]. Symbolic differentiation can be easily automated to be used in scientific codes when restricted to a domain-specific context. The package Theano performs both symbolic and automatic differentiation on expressions defined in Python and generates C++ or CUDA code (to target nVidia GPUs) [6, 9].

The finite element package FEniCS is based around generating local FEM matrix calculations using its Unified Form Language to describe the weak form of the PDE [85, 86]. The finite element shape functions, quadrature rules, and integral calculation are automatically generated and translated into C++ code to be compiled into machine code. The linearization process works symbolically at the coefficient level, i.e. operating on equations after the introduction of the discrete functions of the form $u\left(x\right)=\sum a_i\phi_i\left(x\right)$ where ϕ_i are the basis functions.

1.6 Outline of this work

In Chapter 2, continuum mechanics and electromagnetism are briefly covered, with focus on the treatment material media is covered. In Section 2.8, the quasistatic electric potential formulation that will be the basis for the coupling of the mechanical and electromagnetic fields is developed. One basic problem that can be treated is the deformation of the current-carrying wire in a magnetic field. This problem is solved analytically in Chapter 3.

The finite element beam formulation is developed in Chapter 4 with coupled mechanical, thermal, and electromagnetic fields. The constitutive description and discretization of contacts between beams is discussed in Section 4.3. The implementation of the models and the incorporation of the contact formulation into the FEniCS framework is detailed in Chapter 5. The construction and solution of the numerical system that arrises from the coupled multiphysics finite element formulation is developed and analyzed in Chapter 6. The solution from Chapter 3 is used as one of the benchmarks for the computational model. The behavior of the numerical solution of the dynamic problem is also analyzed, without the assistance of an analytical solution.

Complicated textiles are developed and analyzed in Chapter 7. A framework for using the computational model to compute effective properties of representative volume elements is described. The impact problem illustrated in Figure 1.1.2 is finally solved in Chapter 8, using the beam formulation to model yarns with homogenized properties. Chapter 9 discusses the future direction of this research, with potential applications for the developed framework.

A sample script file for the calculations in Chapter 8 is listed in Appendix A. The quadrature rules for circular cross sections used are listed in Appendix B. Appendix C lists the Butcher Tableaus for the Runge-Kutta methods used. The visualization routine for generating volumetric meshes for beams is described in Appendix D.

Chapter 2

Electromagnetism and Continuum Mechanics

2.1 Introduction

The foundations of continuum mechanics and electromagnetism that are required in the formulation of the fibril model are overviewed. Specifically, the transformation of Maxwell's equations into the rest frame and constitutive theory of moving bodies will be discussed. The reader is directed to Kovetz [72] and Steigmann [107] in particular for the theoretical basis for the coupling continuum mechanics and electromagnetism used throughout this work.

Firstly, the nomenclature used throughout this manuscript is provided in Tables 2.1 and 2.2. Scalar quantities will be regular type face, and typically lower case; vector and tensor quantities will be bold faced, with vectors usually being lower cased and tensors usually being upper case. The choice of units for all calculations is tabulated in Section 2.2. Continuum mechanics and classical electromagnetism are then overviewed in Sections 2.3 and 2.4. The treatment of material responses is discussed in Sections 2.5, 2.6 and 2.7. A quasistatic potential approximation for a conducting body moving through a magnetic field is derived in Section 2.8, and the final system of coupled partial differential equations to be solved is summarized in 2.9.

2.2 Consistent Units

Various unit systems exist for electromagnetism with different benefits; SI units are used for easier coupling with mechanical fields. Standard SI units would be scaled poorly for the small pieces of textiles are to be analyzed. It is necessary to keep the electromagnetic units scaled properly with the mechanical units so that no explicit unit conversions are needed within the computations. That is, the units chosen for each quantity should be consistent. For example, in the standard SI units, $f = qE$ requires consistency between $[N] = [C]\,[V/m]$ and $V = \int Edl$ requires $[V] = \left[\frac{N}{C}\right][m]$. The millimeter, megapascal, newton, and ampere are chosen, and the rest of the units follow from consistency. The units used are detailed in Table 2.3. One interesting note is that the consistent magnetic field strength for these "small" units becomes the kilotesla, which is an extremely large quantity (a rare-earth natural magnet has

Table 2.1: Nomenclature of Operators and Mechanical Quantities

t	Time (s)
$\mathbf{e}_i$	Coordinate orthonormal basis in spatial configuration
$\mathbf{E}_i$	Coordinate basis in reference configuration
$\mathbf{x}$	Laboratory frame spatial variable in orthogonal basis; Material position in current configuration (m)
$\mathbf{X}$	Material rest frame spatial variable in curvilinear basis (reference configuration) (m)
$\partial\Omega$	Boundary surface of volume Ω, or boundary contour of open surface Ω
$\Omega \cup \Psi$	Union of two regions; all points in either regions
$\Omega \cap \Psi$	Intersection of two regions, or part of a surface contained in a body; all points in both regions
$\Omega \backslash \Psi$	Subtraction of two regions; all points in Ω but not Ψ.
$\dot{()}, \frac{d()}{dt}$	Time derivative with respect to the material frame
$\left.\frac{\partial()}{\partial t}\right\rvert_{\mathbf{x}}$	Time derivative with respect to a fixed coordinate in frame $\mathbf{x}$
$\nabla_x, \frac{\partial}{\partial \mathbf{x}}$	Gradient operator, nabla, with respect to coordinate frame $\mathbf{x}$
$[[\mathbf{a}]]$	Jump of quantity $\mathbf{a}$ across a surface of discontinuity, $\mathbf{a}^+ - \mathbf{a}^-$ $(= 0$ if the field is continuous). The unit normal $\mathbf{n}$ of the surface points towards the $+$ side.
$\overset{*}{()}$	Flux derivative w.r.t. a moving frame with velocity $\mathbf{v}$, $\overset{*}{\mathbf{X}} = \frac{\partial \mathbf{X}}{\partial t} + (\nabla \cdot \mathbf{X})\,\mathbf{v} - \nabla \times (\mathbf{v} \times \mathbf{X})$
$\Vert x \Vert_n$	The nth norm of the quantity x
$\lvert x \rvert$	Magnitude of a vector (2-norm), determinant of a matrix, or volume of a region.
$\mathbf{a} \cdot \mathbf{b}$	Inner (dot) product, $a_i b_i$
$\mathbf{A}{:}\mathbf{B}$	Double inner product, $A_{ij}B_{ij}$
$\mathbf{a} \otimes \mathbf{b}$	Tensor outer product, $a_i b_j \mathbf{e}_i \otimes \mathbf{e}_j$
ρ	Mass density $({}^{kg}/_{m^3})$
ε	Specific energy $({}^{J}/_{kg})$
T	Temperature (K)
$\mathbf{T}$	Cauchy stress tensor (current configuration)$({}^{N}/_{m^2})$
$\mathbf{P}$	First Piola-Kirchhoff stress tensor (two-footed) $({}^{N}/_{m^2})$
$\mathbf{S}$	Second Piola-Kirchhoff stress tensor (reference configuration) $({}^{N}/_{m^2})$

Table 2.2: Nomenclature of Electromagnetic Quantities

Q	Total charge (C)
q	Charge density $({}^{C}\!/_{m^{3}})$
$\mathbf{J}$	Current density $({}^{A}\!/_{m^{2}})$
σ	Surface charge density
$\mathbf{k}$	Surface current density
$\mathbf{E}$	Electric field $({}^{V}\!/_{m})$
$\mathbf{B}$	Magnetic flux density (T)
$\mathbf{D}$	Electric displacement field $({}^{C}\!/_{m^{2}})$
$\mathbf{H}$	Magnetizing field $({}^{A}\!/_{m})$
$\mathbf{P}$	Electric polarization $({}^{C}\!/_{m^{2}})$
$\mathbf{M}$	Magnetization $({}^{A}\!/_{m})$
$\boldsymbol{\mathcal{J}}$	Conduction current density $({}^{A}\!/_{m^{2}})$
$\boldsymbol{\mathcal{E}}$	Electromotive intensity $({}^{V}\!/_{m})$
$\boldsymbol{\mathcal{H}}$	Magnetomotive intensity $({}^{A}\!/_{m})$
$\boldsymbol{\mathcal{M}}$	Lorentz Magnetization $({}^{A}\!/_{m})$
V	Electric potential (V)
$\mathbf{A}$	Magnetic vector potential $({}^{Vs}\!/_{m})$
ϵ_0	Permittivity of Free space
μ_0	Permeability of free space
$\boldsymbol{\chi}$	Relative Susceptibility
ϵ_r	Relative Permittivity
ϵ_{0r}	Relative Permittivity *in the reference configuration.*

Table 2.3: Units

	Used Unit	Conversion	Standard SI
Length	mm	$10^{-3}m$	m
Time	ms	$10^{-3}s$	s
Force	N	$1N$	N
Temperature	K	$1K$	K
Mass	g	$10^{-3}kg$	kg
Pressure, Stiffness	MPa	$10^{6}Pa$	$\frac{N}{m^2}$
Current, I	A	$1A$	A
Charge, Q	mC	$10^{-3}sA$	C, sA
Electric Field Strength, E, $\mathcal{E}$	$\frac{V}{mm}$, $\frac{N}{mC}$	$10^{3}\frac{V}{m}$	V/m, $\frac{N}{C}$
Magnetic Field Strength, B	kT	$10^{3}T$	T, $\frac{kg}{s^2A}$
Electric Potential, V	V	$1V$	V, $\frac{kgm^2}{s^3A}$
Electric Conductivity, σ	$\frac{S}{mm}$	$10^{-3}\frac{S}{m}$	$\frac{S}{m}$, $\frac{A}{Vm}$
Electric Contact Resistance, σ_C	$\frac{A}{mm^2V}$	$10^{6}\frac{A}{m^2V}$	$\frac{A}{m^2V}$
Power	W	$1W$	W, $\frac{kgm^2}{s^3}$
Thermal Conductivity, k	$\frac{W}{mmK}$ $\frac{gmm}{ms^3K}$	$10^{3}\frac{W}{mk}$	$\frac{W}{mK}$, $\frac{kgm}{s^3K}$
Heat transfer coefficient, h	$\frac{W}{mm^2K}$	$10^{6}\frac{W}{m^2K}$	$\frac{W}{m^2K}$

a strength of about $1T$, for comparison.)

2.3 Mechanics

The current position of each particle point inside of a material located at a reference position $\mathbf{X}$ is sought as a function of time. The path can be expressed as either the position function $\mathbf{x}$, or the displacement function $\mathbf{u}$, related by

$$\mathbf{x}(t, \mathbf{X}) = \mathbf{X} + \mathbf{u}(t, \mathbf{X}).$$

The initial state of the material is not necessarily the reference configuration, i.e. $\mathbf{x}(t = 0) \neq \mathbf{X}$, since a deformed state may be used to begin the analysis (of particular interest, consider a woven textile where the fibrils are already deformed at the beginning of a simulation.) It is also sometimes useful to define a configuration that is not the material's stress-free state. It will be convenient in the beam analysis developed in Chapter 4 when the beam is naturally curved. Further configurations may also be used when constitutive models for damage are utilized, but that will not be treated in this work. The following nomenclature for coordinate frames will be used throughout this work:

- $\boldsymbol{\xi}$ is the ideal configuration, describing an ideal state for defining material points that will be useful later for the assumed deformation function of the thin fibers in question;
- $\mathbf{x}(\boldsymbol{\xi}, t)$ is the current configuration, the present position of material point in the body;
- $\mathbf{x}(\boldsymbol{\xi}, t = 0)$ is the initial configuration, where the material point was at the beginning of the calculation; and
- $\mathbf{X}_0(\boldsymbol{\xi})$ is the reference, or natural, configuration, where the material point was when the body "came out of the factory" and had no plastic or elastic deformation.

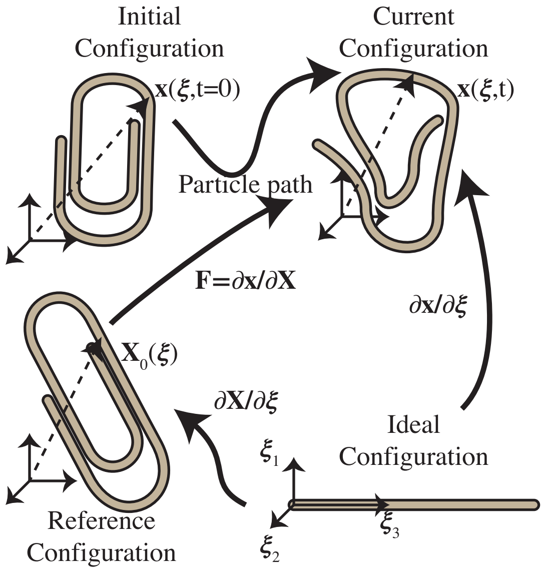


Figure 2.3.1: Configurations for describing finite deformation problems

In many problems in solid mechanics, the reference configuration is used to define material points, so all of the fields would be expressed as functions of $\mathbf{X}_0$. The various configurations are illustrated in Figure 2.3.1.

The deformation gradient of the motion is defined as

$$\mathbf{F} = \nabla_X \mathbf{x} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \mathbf{I} + \nabla_X \mathbf{u}$$

and can be calculated in terms of either the material coordinates or the displacement field. The determinant of the deformation gradient is often used,

$$J = \det \mathbf{F}$$

and is referred to in this context as the Jacobian of the deformation (though in the mathematics terminology $\mathbf{F}$ is the Jacobian.) Strain measures that will be used later are the right Cauchy-Green strain,

$$\mathbf{C} = \mathbf{F}^T\mathbf{F}$$

and the Green-Lagrange strain tensor,

$$\mathbf{E} = \frac{1}{2}\left(\mathbf{F}^T\mathbf{F} - \mathbf{I}\right).$$

Three stress measures are used: the Cauchy stress, $\mathbf{T}$, the first Piola-Kirchhoff stress, $\mathbf{P}$, and the second Piola-Kirchhoff stress, $\mathbf{S}$. Two possible definitions of the stress tensors are possible which are transposes of each other. The following convention is used, where the tensors are defined to operate on their respective unit normals, $\mathbf{n}$ in the current and $\mathbf{N}$ in the reference configuration, to produce the traction in the spatial configuration,$\mathbf{t}$, as so:

$$\mathbf{t}da = \mathbf{T}\mathbf{n}da = \mathbf{P}\mathbf{N}dA.$$

The fundamental physical laws that need to be solved in mechanics are the balance of mass, energy, and linear momentum. In the spatial configuration, these laws are, respectively,

$$\begin{aligned}\dot{\rho} &= -\rho\nabla_x \cdot \mathbf{v} \\ \rho\dot{\varepsilon} &= \mathbf{T} : \nabla_x\mathbf{v} - \nabla_x \cdot \mathbf{q} + r \\ \rho\dot{\mathbf{v}} &= \nabla_x \cdot \mathbf{T} + \mathbf{f}.\end{aligned}$$

For a solid, the mass continuity equation simplifies to $\rho = (\det \mathbf{F})\,\rho_0 = J\rho_0$. The balance of angular momentum is also required but is easily satisfied by the conditions

$$\begin{aligned}\mathbf{T} &= \mathbf{T}^T \\ \mathbf{FP} &= \mathbf{P}^T\mathbf{F}^T \\ \mathbf{S} &= \mathbf{S}^T\end{aligned}$$

on the symmetries of the stress tensors. The force density is sometimes used as having units of force per mass, so that the right hand side of the balance of linear momentum is instead $\nabla_x \cdot \mathbf{T} + \rho\mathbf{b}$. The form of force per volume will be more convenient when coupling with electromagnetic fields, whose forces are not associated with the mass density of the material, but instead with the charge and current densities. Similarly, the heat generation term is also sometimes given per unit mass, but the heat generation per unit volume will be more applicable for electromagnetic sources.

2.4 Maxwell's Equations

Here only the macroscopic fields are considered. The derivation of the macroscopic fields from the presence of material media can be found in [63, 72]. Similar to continuum mechanics, each field is a spatial average of a microscopic structure. The first equation is the principal of charge conservation, which in its local form as a partial differential equation is

$$\frac{\partial q}{\partial t} + \nabla_x \cdot \mathbf{J} = 0, \tag{2.4.1}$$

or, as an integral equation, is

$$\frac{\partial Q}{\partial t} + \oint_{\partial\Omega} \mathbf{J} \cdot \mathbf{n} d\Gamma = 0$$

where Q is the total charge in the domain. The total charge in a domain is generally the total of the charge densities, surface charges, and discrete point charges,

$$Q = \int_{\Omega} q d\Omega + \sum_i \int_{S_i} \sigma dS + \sum_{i\in\Omega} Q_i.$$

Surface containing surface charges are denoted by S_i. The above local form only applies when the charge is sufficiently smooth and only composes of a charge density field.

As vector partial differential equations, Maxwell's equations are

$$\begin{aligned}
\nabla_x \times \mathbf{E} &= -\frac{\partial \mathbf{B}}{\partial t} && (2.4.2)\\
\nabla_x \cdot \mathbf{B} &= 0 && (2.4.3)\\
\nabla_x \cdot \mathbf{D} &= q && (2.4.4)\\
\nabla_x \times \mathbf{H} &= \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}. && (2.4.5)
\end{aligned}$$

The corresponding integral forms are

$$\begin{aligned}
\oint_{\partial A} \mathbf{E} \cdot d\mathbf{l} &= -\frac{d}{dt}\int_A \mathbf{B} \cdot \mathbf{n} dA\\
\oint_{\Omega} \mathbf{B} \cdot \mathbf{n} dA &= 0\\
\oint_{\Omega} \mathbf{D} \cdot \mathbf{n} dA &= Q\\
\oint_{\partial A} \mathbf{H} \cdot d\mathbf{l} &= I + \frac{d}{dt}\int \mathbf{D} \cdot \mathbf{n} dA
\end{aligned}$$

where I is the total current passing through the area A. Similarly to the total charge, the total current can be made of a current density and surface currents,

$$I = \int_A \mathbf{J} \cdot \mathbf{n} dA + \sum_{S_i} \int_{S_i \cap A} \mathbf{k} \cdot \mathbf{n} dl,$$

where discrete line-currents are also possible but were neglected.

The fields $\mathbf{E}$ and $\mathbf{B}$ are paired together as different components of the same four-dimensional tensor field, as are $\mathbf{D}$ and $\mathbf{H}$; however, the covariant formulation is beyond the scope of this section. In a vacuum, these two pairs are related together by the aether relations,

$$\begin{aligned}
\mathbf{D} &= \epsilon_0 \mathbf{E}\\
\mathbf{H} &= \mu_0^{-1} \mathbf{B}.
\end{aligned}$$

In the presence of a material medium, additional polarization and magnetization fields are present that contribute to the displacement and magnetizing fields. In this case, the relations between the two pairs of fields are

$$\begin{aligned}
\mathbf{D} &= \epsilon_0 \mathbf{E} + \mathbf{P}\\
\mathbf{H} &= \mu_0^{-1} \mathbf{B} - \mathbf{M}.
\end{aligned}$$

This work uses the interpretation that the aether relations are always present—both in the vacuum and in a material—and constitutive responses for media only appear in the polarization and magnetization fields. To further this interpretation, the vacuum can be interpreted to have the constitutive law $\mathbf{P} = \mathbf{0}$ and $\mathbf{M} = \mathbf{0}$.

In this work the potential formulation will be used extensively. The magnetic vector potential $\mathbf{A}$ and electric scalar potential V can be constructed so that

$$\begin{aligned}\mathbf{B} &= \nabla_x \times \mathbf{A} \\ \mathbf{E} &= -\nabla_x V - \frac{\partial \mathbf{A}}{\partial t}.\end{aligned}$$

This formulation automatically satisfies Equations 2.4.5 and 2.4.5 because the differential operators $\nabla \cdot \nabla\times$ and $\nabla \times \nabla$ are zero.

2.5 Material Frame Invariance

The fundamental laws of mechanics, as well as any constitutive response laws developed, are required to be invariant under rigid body (Galilean) transformations: i.e. the balance laws on a body should be valid in any inertial frame. However, the Aether relations are only invariant with respect to Lorentz transformations, of which Galilean transformations are the non-relativistic limit. The Aether relations are posited to hold in some laboratory frame, and do not hold in an arbitrary material frame. Using a relativistic formulation of continuum mechanics would satisfy this issue, but taking care to switch between the fixed laboratory frame and the material frame is much easier, and maintains compatibility with the many years of continuum mechanics theory. To formulate constitutive responses between the electromagnetic fields and a material body, the transformation of those fields onto the body's rest frame is required.

The Galilean transformation from frame x to x^+ has the form, stated using Einstein summation convention,

$$x_i^+ = A_{ij}\left(x_j - u_j t\right)$$

where $\mathbf{X}$ is a constant coordinate shift, $\mathbf{A}$ is a coordinate rotation, and $\mathbf{u}$ is the velocity of the transformations. The electromagnetic fields transform as

$$\begin{aligned}\mathbf{E}^+ &= \mathbf{E} + \mathbf{u} \times \mathbf{B} \\ \mathbf{B}^+ &= \mathbf{B} \\ \mathbf{D}^+ &= \mathbf{D} \\ \mathbf{H}^+ &= \mathbf{H} - \mathbf{u} \times \mathbf{D}.\end{aligned}$$

The polarization and magnetization densities transform as their counter parts,

$$\begin{aligned}\mathbf{P}^+ &= \mathbf{P} \\ \mathbf{M}^+ &= \mathbf{M} + \mathbf{u} \times \mathbf{D}.\end{aligned}$$

A charge moving with a material will be yield a current equal to $\mathbf{J} = q\mathbf{v}$ when observed in the lab frame, but since it is not moving relative to the material, no current is observed in the material rest frame. Thus, when the frame is moving relative to a charge and current density, the current density transforms as

$$\mathbf{J}^+ = \mathbf{J} - q\mathbf{u}.$$

The invariant quantities are the conduction current density, $\boldsymbol{\mathcal{J}}$, the electromotive intensity, $\boldsymbol{\mathcal{E}}$, the magnetomotive intensity, $\boldsymbol{\mathcal{H}}$, and the Lorentz magnetization, $\boldsymbol{\mathcal{M}}$:

$$\begin{aligned}\boldsymbol{\mathcal{J}} &= \mathbf{J} - q\mathbf{v}\\ \boldsymbol{\mathcal{E}} &= \mathbf{E} + \mathbf{v} \times \mathbf{B}\\ \boldsymbol{\mathcal{H}} &= \mathbf{H} - \mathbf{v} \times \mathbf{D}\\ \boldsymbol{\mathcal{M}} &= \mathbf{M} + \mathbf{v} \times \mathbf{P}.\end{aligned}$$

The necessary Maxwell's equations can be rewritten in terms of these invariants,

$$\begin{aligned}\nabla_x \times \boldsymbol{\mathcal{E}} &= \overset{*}{\mathbf{B}}\\ \nabla_x \times \boldsymbol{\mathcal{H}} &= \boldsymbol{\mathcal{J}} + \overset{*}{\mathbf{D}}\end{aligned}$$

where $\overset{*}{\mathbf{X}}$ denotes the flux derivative, $\overset{*}{\mathbf{X}} = \frac{\partial \mathbf{X}}{\partial t} + (\nabla_x \cdot \mathbf{X})\,\mathbf{v} - \nabla_x \times (\mathbf{v} \times \mathbf{X})$ in differential notation. It satisfies the property for moving surfaces $\frac{d}{dt}\int_{A(t)} \mathbf{X} \cdot \mathbf{n} dA = \int_{A(t)} \overset{*}{\mathbf{X}} \cdot \mathbf{n} dA$.

2.6 Constitutive Equations

A Neohookean hyperelastic constitutive law is used for the Helmholtz strain energy density,

$$\psi = \frac{\mu}{2}\left(\mathrm{tr}\mathbf{C} - 3\right) - \mu \log J + \frac{\lambda}{2}\left(\log J\right)^2.$$

The Lamé parameters μ and λ can be calculated from the Young's modulus E and Poisson ratio ν by

$$\begin{aligned}\mu &= \frac{E}{2\left(1+\nu\right)}\\ \lambda &= \frac{E\nu}{\left(1+\nu\right)\left(1-2\nu\right)}.\end{aligned}$$

The second Piola-Kirchhoff stress is obtained as the derivative of the Helmholtz strain energy density with respect to the right Cauchy Green strain tensor,

$$\mathbf{S} = \frac{\partial \psi}{\partial \mathbf{C}}.$$

For the Neohookean constitutive law, this yields a form for the stress

$$\mathbf{S} = \mu \mathbf{I} + \left(\lambda \log J - \mu\right) \mathbf{C}^{-T}.$$

By the chain rule, the following relations between to the strain energy to variational displacement fields $\delta\mathbf{u}$ are all the same:

$$\delta\mathbf{u} \cdot \frac{\partial \psi}{\partial \mathbf{u}} = \left(\nabla_X \delta\mathbf{u}\right) : \frac{\partial \psi}{\partial \mathbf{F}} = \left(\nabla_X \left(\delta\mathbf{u}\right)\mathbf{F}\right) : \frac{\partial \psi}{\partial \mathbf{C}}.$$

These relationships will be exploited in the implementation, where the Gâteaux derivative with respect to test functions on the strain energy will be used instead of the stress tensor itself.

The internal stored energy has three components: the heat capacitance, the strain energy, and energy stored in electromagnetic effects. The constitutive response is

$$\rho_0 \varepsilon = w_{thermal} + w_{strain} + w_{EM} = \rho_0 c_p T + \psi + w_{EM}.$$

The electromagnetic stored energy will be neglected in this work due to the nature of the assumptions used. Considering only the thermal and mechanical components, the rate of change of energy is

$$\rho_0 \dot{\varepsilon} = \rho_0 c_p \dot{T} + \dot{\psi}.$$

The strain energy used only depends on the deformation and temperature, i.e. $\psi = \psi\left(\mathbf{u}, T\right)$. The material derivative of the strain energy can be manipulated by the chain rule to obtain

$$\dot{\psi} = \frac{\partial \psi}{\partial \mathbf{u}} \cdot \frac{d\mathbf{u}}{dt} + \frac{\partial \psi}{\partial T}\frac{dT}{dt} = \frac{\partial \psi}{\partial \mathbf{F}} : \frac{d\mathbf{F}}{dt} + \frac{\partial \psi}{\partial T}\frac{dT}{dt}. \tag{2.6.1}$$

The term $\frac{\partial \psi}{\partial \mathbf{F}} : \dot{\mathbf{F}}$ cancels out the stress power term $\mathbf{P} : \nabla_X \mathbf{v}$ in the balance of energy. Responses for ψ taking into account electromagnetic fields are possible and would need to be taken into account in Equation 2.6.1. The derivative with respect to temperature of the above Neohookean strain energy law is

$$\frac{\partial \psi}{\partial T} = \frac{\partial \mu}{\partial T}\left(\frac{1}{2}\left(\mathrm{tr}\mathbf{C} - 3\right) - \log J\right) + \frac{\partial \lambda}{\partial T}\frac{1}{2}\left(\log J\right)^2.$$

Fourier's law couples the heat flux vector to the temperature gradient,

$$\mathbf{q} = -\mathbf{k}\nabla_x T$$

where $\mathbf{k}$ is the thermal conductivity tensor. Isotropy in the reference configuration will be assumed, so that it can be treated as a scalar (or a diagonal tensor, $\mathbf{k}_0 = k\mathbf{I}$). Using the above constitutive responses, the energy balance equation is simplified to

$$\left(\rho_0 c_p + \frac{\partial \psi}{\partial T}\right)\dot{T} = \nabla_X \cdot \mathbf{k}_0 \nabla_X T + r.$$

More care must be taken for electromagnetic responses. Constitutive laws must be written in the rest frame of a material, so the Galilean invariants are used. The polarization of a medium is $\mathbf{P}' = \epsilon_0 \boldsymbol{\chi} \boldsymbol{\mathcal{E}}$ where $\boldsymbol{\chi}$ is the susceptibility of the material and ϵ_0 is the free space permittivity. The displacement field inside of a polarizable media moving through a magnetic field can be calculated by

$$\mathbf{D} = \epsilon_0 \mathbf{E} + \mathbf{P} = \epsilon_0 \mathbf{E} + \epsilon_0 \boldsymbol{\chi} \boldsymbol{\mathcal{E}} = \epsilon_r \boldsymbol{\mathcal{E}} - \epsilon_0 \mathbf{v} \times \mathbf{B}. \tag{2.6.2}$$

Similarly, Ohm's law for a conductor,

$$\boldsymbol{\mathcal{J}} = \boldsymbol{\sigma} \boldsymbol{\mathcal{E}} \tag{2.6.3}$$

where $\boldsymbol{\sigma}$ is the electrical conductivity tensor, holds in the material's rest frame (see [63] p572; [72] p86,135). As with the thermal conductivity, only isotropic materials are considered, so it will be treated as a scalar. The Seebeck effect will be briefly considered for the purpose of adding a coupling to the system in Chapter 6. The effect induces an electromotive force in response to temperature gradients, so that the constitutive response for the material is

$$\mathcal{J} = \boldsymbol{\sigma}\left(\boldsymbol{\mathcal{E}} - S\nabla_x T\right)$$

where S is the Seebeck coefficient that can be either positive or negative, depending on the material. The reader is directed to Kovetz [72] for a formal development of electromagnetic material constitutive responses.

2.7 Electromagnetic Forces and Sources

A moving particle observes a force dependent on the electric field in its rest frame, $\mathbf{F} = Q\boldsymbol{\mathcal{E}}$, which has the familiar form $\mathbf{F} = Q\left(\mathbf{E} + \mathbf{v} \times \mathbf{B}\right)$ when calculated in the laboratory frame. The charge and current densities within a continuous media yield a force density of

$$\mathbf{f} = q\mathbf{E} + \mathbf{J} \times \mathbf{B}$$

or, as stated in the material rest frame,

$$\mathbf{f} = q\boldsymbol{\mathcal{E}} + \boldsymbol{\mathcal{J}} \times \mathbf{B}.$$

These body forces provide the coupling from the electromagnetic fields to the mechanical fields.

Conducting materials also exhibit Joule heating,

$$r = \boldsymbol{\mathcal{J}} \cdot \boldsymbol{\mathcal{E}} = \boldsymbol{\mathcal{E}} \cdot \boldsymbol{\sigma}\boldsymbol{\mathcal{E}}.$$

2.8 Quasistatic Potential Approximation in the Presence of Moving Conducting Media

2.8.1 Assumptions

This work focuses on the scenario of electric conductors moving within an externally applied magnetic field, e.g. when a strong electromagnet is near the conductors. As discussed in the previous chapter, solving Maxwell's equations is infeasible. The electromagnetic problem is simplified and rephrased in terms of the electric potential. The following assumptions are made in developing the formulation:

1. The magnetic B field is dominated by an externally applied field; the contribution by the currents is negligible.
2. The average time derivative of the E field is zero on mechanical time scales $\left\langle \frac{\partial}{\partial t} E \right\rangle \Big|_{\delta t} = 0$
   (a) Similarly, the B field is set to a constant; a gradually changing B field could be applied by using $\left\langle \frac{\partial}{\partial t} B \right\rangle \Big|_{\delta t} = \frac{\partial B}{\partial t}\Big|_{applied}$.

With a static external magnetic field, $\frac{\partial \mathbf{B}}{\partial t} = 0$, the magnetic vector potential will also be constant, $\frac{\partial \mathbf{A}}{\partial t} = 0$.

2.8.2 Partial Differential Equation

The electric potential is not invariant, but the invariant quantity $V' = V - \mathbf{v} \cdot \mathbf{A}$ is unwieldy for this purpose due to its dependence of the magnetic vector potential. Let V be the *Lab Frame* electric potential so that the electric field $\mathbf{E}$ then satisfies $\mathbf{E} = -\nabla_x V$. Faraday's law is trivially satisfied by this formulation, since the curl of the gradient is zero, and Gauss's law for magnetism forms a constraint on the applied $\mathbf{B}$ that can be chosen. Three equations remain, namely Equations 2.4.4,2.4.5, and 2.4.1. One is redundant, as charge conservation can be obtained by combining the remaining two. Because magnetizing fields are not of interest—and curl-type differential equations are a hassle to work with—the two equations to be dealt with are Gauss's law and charge conservation. In these two equations, there are two unknowns: the electric potential V and the charge q.

Charge conservation can be written in terms of the conduction current density,

$$\begin{aligned} -\frac{\partial q}{\partial t} &= \nabla_x \cdot \mathbf{j} \\ &= \nabla_x \cdot \boldsymbol{\mathcal{J}} + \nabla_x \cdot q\mathbf{v} \end{aligned}$$

Inserting Ohm's Law and then inserting the lab frame potential,

$$\begin{aligned} -\frac{\partial q}{\partial t} &= \nabla_x \cdot \sigma \boldsymbol{\mathcal{E}} + \nabla_x \cdot q\mathbf{v} \\ &= -\nabla_x \cdot \sigma \nabla_x V + \nabla_x \sigma \mathbf{v} \times \mathbf{B} + \nabla_x \cdot q\mathbf{v}. \end{aligned}$$

Now to proceed with Gauss's law,

$$\begin{aligned} q &= \nabla_x \cdot \mathbf{D} \\ &= \nabla_x \cdot (\epsilon_0 \mathbf{E} + \mathbf{P}) \\ &= \nabla_x \cdot (\epsilon_0 (1+\chi) \mathbf{E} + \epsilon_0 \chi \mathbf{v} \times \mathbf{B}), \end{aligned}$$

which can be written terms of the potential,

$$q = -\nabla_x \cdot \epsilon_r \nabla_x V + \nabla_x \cdot \epsilon_0 \chi \mathbf{v} \times \mathbf{B}.$$

Multiplying by $\frac{\sigma}{\epsilon_r}$,

$$\nabla_x \cdot \sigma \nabla_x V = -\frac{\sigma}{\epsilon_r} q + \nabla_x \cdot \sigma \frac{\chi}{1+\chi} \mathbf{v} \times \mathbf{B}.$$

This result can be substituted back into the continuity equation to obtain an equation in terms of the charge density only,

$$-\frac{\partial q}{\partial t} = \frac{\sigma}{\epsilon_r} q - \nabla_x \cdot \sigma \frac{\chi}{1+\chi} \mathbf{v} \times \mathbf{B} + \nabla_x \sigma \mathbf{v} \times \mathbf{B} + \nabla_x \cdot q\mathbf{v}$$

and simplified as

$$\begin{aligned} -\frac{\partial q}{\partial t} - \nabla_x \cdot q\mathbf{v} &= \frac{\sigma}{\epsilon_r} q - \nabla_x \cdot \sigma \left(\frac{\chi}{1+\chi} - 1 \right) \mathbf{v} \times \mathbf{B} \\ -\frac{\partial q}{\partial t} - \nabla_x \cdot q\mathbf{v} &= \frac{\sigma}{\epsilon_r} q + \nabla_x \cdot \sigma \left(\frac{1}{1+\chi} \right) \mathbf{v} \times \mathbf{B} \end{aligned}$$

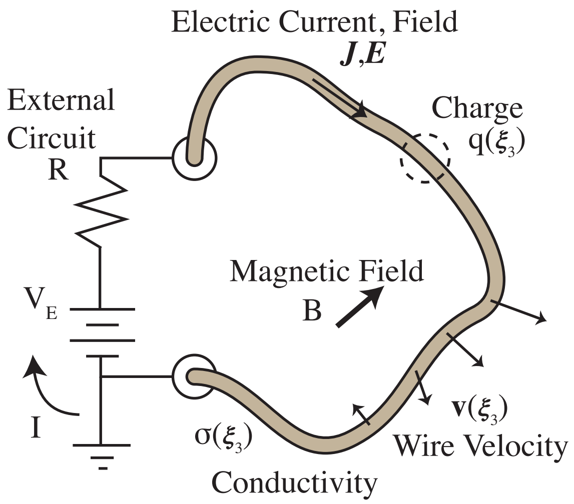


Figure 2.8.1: A wire moving through a magnetic field.

yielding

$$-\frac{\epsilon_r}{\sigma}\left(\frac{\partial q}{\partial t}+\nabla_x\cdot q\mathbf{v}\right)=\nabla_x\cdot\epsilon_r\left(\frac{1}{1+\chi}\right)\mathbf{v}\times\mathbf{B}+q.$$

The term $\frac{\partial q}{\partial t}+\nabla_x\cdot q\mathbf{v}$ is the derivative of the moving charge carriers with respect to the motion of the material. The permittivity of free space ϵ_0 is approximately equal to $8.85\times10^{-12}\frac{F}{m}$ and the conductivity σ of conductors is on the order of $10^6\frac{S}{m}$. The ratio $\frac{\epsilon_r}{\sigma}$ is extremely small and represents a very stiff time constant for the relaxation time of free charges. Taking the limit as $\frac{\epsilon_r}{\sigma}\to0$, the steady state result can be used as an approximation. The charge in the material will then satisfy the relation

$$q=-\nabla_x\cdot\epsilon_0\mathbf{v}\times\mathbf{B} \tag{2.8.1}$$

Using this result in Gauss's law above,

$$-\nabla_x\cdot\epsilon_0\mathbf{v}\times\mathbf{B}=-\nabla_x\cdot\epsilon_r\nabla_xV+\nabla_x\cdot\epsilon_0\chi\mathbf{v}\times\mathbf{B}$$

yields an equation for the potential, as measured in the stationary lab frame, in the presence of a moving conducting medium:

$$\nabla_x\cdot\epsilon_r\nabla_xV=\nabla_x\cdot\epsilon_r\mathbf{v}\times\mathbf{B}.$$

2.8.3 Boundary Conditions

It is desirable to have boundary conditions that would yield a back electromotive force effect, as is observed in an electric motor, for example. The simple case in Figure 2.8.1 will be used in this work. Using two voltage boundary conditions has current-sourcing effects that are not intuitive. One side of the wire is set to ground to pin the constant mode in the

PDE, and on the other side, a mixed boundary condition is used that couples the body to an the external circuit.

The total current leaving the body and entering the circuit is $I = \int_\Gamma \boldsymbol{\mathcal{J}} \cdot \mathbf{n} d\Gamma$. Assuming $V(0) = 0$, the body has to match up with the circuit by the integral relation

$$V(L) = -R \int_\Gamma \boldsymbol{\mathcal{J}}(L) \cdot \mathbf{n} d\Gamma + \Delta V_E.$$

Assuming that there is an ideal electrical connection at the surface, with the area of the surface being $A = |\Gamma|$, the above relation can be phrased in terms of the current density as

$$\mathcal{J}_n|_\Gamma = -\underbrace{\frac{1}{RA}}_{\bar{r}} V|_\Gamma + \underbrace{\frac{\Delta V_E}{RA}}_{\bar{J}}, \tag{2.8.2}$$

thus giving a mixed boundary condition linearly relating the normal current density to the potential with coefficients $\bar{r}$ and $\bar{J}$. Coupling to more complicated circuits requires solving an additional Kirchhoff's law network system while solving the PDE.

2.9 Summary of Equations

The three coupled partial differential equations derived can be summarized by

$$\begin{aligned} \rho \ddot{\mathbf{x}} &= \nabla_x \cdot \mathbf{T} + q \boldsymbol{\mathcal{E}} + \boldsymbol{\mathcal{J}} \times \mathbf{B} + \mathbf{f} \\ \rho c_p \dot{T} + \frac{\rho}{\rho_0} \dot{\psi} &= \nabla_x \cdot \mathbf{k} \nabla_x T + \mathbf{T} : \nabla_x \mathbf{v} + \boldsymbol{\mathcal{J}} \cdot \boldsymbol{\mathcal{E}} + r \\ 0 &= \nabla_x \cdot \epsilon_r \nabla_x V - \nabla_x \cdot \epsilon_r \mathbf{v} \times \mathbf{B}. \end{aligned}$$

where $\mathbf{f}$ and r are force and heat generation densities that are not electromagnetic in nature. These equations are all stated in the current configuration, and it is often easier to use the reference or ideal configuration to perform these calculations.

To push back these equations into the reference and ideal configurations, note that the divergence of some vector transforms as $\nabla_x \cdot \mathbf{a} = \nabla_X \cdot \mathbf{F}^{-1} \mathbf{a} = \nabla_\xi \cdot \left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \right)^{-1} \mathbf{a}$ and its gradient transforms as $\nabla_x \mathbf{a} = (\nabla_X \mathbf{a}) \mathbf{F}^{-1} = (\nabla_\xi \mathbf{a}) \left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}} \right)^{-1}$.[1]

The material properties $\mathbf{k}$, $\boldsymbol{\sigma}$ and $\boldsymbol{\chi}$ are evaluated in the spatial configuration, after the material has been stretched. Let $\mathbf{k}_0$, $\boldsymbol{\sigma}_0$ and $\boldsymbol{\chi}_0$ be the properties of the material in its reference configuration. Using the differential forms above, it can be determined that these

[1] This work tries to avoid index notation, but it is necessary to verify these transformations: using Einstein summation convention, the divergence can be manipulated as $\frac{\partial}{\partial x_i} a_i = \frac{\partial}{\partial X_i} \frac{\partial X_i}{\partial x_j} a_j = \frac{\partial}{\partial \xi_i} \frac{\partial \xi_i}{\partial x_j} a_j$ and the gradient can be manipulated as $\frac{\partial a_i}{\partial x_j} = \frac{\partial a_i}{\partial X_k} \frac{\partial X_k}{\partial x_j} = \frac{\partial a_i}{\partial \xi_k} \frac{\partial \xi_k}{\partial x_j}$, where the inverses of the transformation gradients are identified as $\left[\mathbf{F}^{-1} \right]_{ij} = \frac{\partial X_i}{\partial x_j}$.

material properties must transform in response to material deformation as

$$\begin{aligned}
\mathbf{k} &= \frac{1}{J}\mathbf{F}\mathbf{k}_0\mathbf{F}^T \\
\boldsymbol{\sigma} &= \frac{1}{J}\mathbf{F}\boldsymbol{\sigma}_0\mathbf{F}^T \\
\boldsymbol{\chi} &= \frac{1}{J}\mathbf{F}\boldsymbol{\chi}_0\mathbf{F}^T.
\end{aligned}$$

As observed, a material once isotropic is not necessarily isotropic after it has been deformed. The relative tensor permittivity is not a material property due to the aether relation of free space. Thus, as the material deforms, the relative permittivity changes as

$$\boldsymbol{\epsilon}_r = \epsilon_0\mathbf{I} + \epsilon_0\boldsymbol{\chi} = \epsilon_0\mathbf{I} + \epsilon_0\frac{1}{J}\mathbf{F}\boldsymbol{\chi}_0\mathbf{F}^T.$$

Applying these transformation rules, all of the derivatives can be made with respect to the the reference configuration,

$$\begin{aligned}
\rho_0\ddot{\mathbf{x}} &= \nabla_X \cdot \mathbf{P} + \frac{\rho_0}{\rho}\left(q\boldsymbol{\mathcal{E}} + \boldsymbol{\mathcal{J}} \times \mathbf{B} + \mathbf{f}\right) \\
\left(\rho_0 c_p + \frac{\partial\psi}{\partial T}\right)\dot{T} &= \nabla_X \cdot \mathbf{k}_0\nabla_X T + \frac{\rho_0}{\rho}\left(\boldsymbol{\mathcal{J}} \cdot \boldsymbol{\mathcal{E}} + r\right) \\
0 &= \nabla_X \cdot \epsilon_0\left(J\mathbf{F}^{-1}\mathbf{F}^{-T} + \boldsymbol{\chi}_0\right)\nabla_X V - \nabla_X \cdot \epsilon_0\left(J\mathbf{F}^{-1}\mathbf{F}^{-T} + \boldsymbol{\chi}_0\right)\mathbf{F}^T\mathbf{v} \times \mathbf{B}.
\end{aligned}$$

Note that in the potential equation, the permittivity tensor has both the reference tensor susceptibility as well as a push back of free space into the reference configuration. The conduction current density can be determined from the potential field by

$$\boldsymbol{\mathcal{J}} = \frac{1}{J}\mathbf{F}\sigma_0\mathbf{F}^T\left(\underbrace{-\mathbf{F}^{-T}\nabla_X V + \mathbf{v} \times \mathbf{B}}_{=\boldsymbol{\mathcal{E}}}\right).$$

Chapter 3

Analytical Solution for the Magnetically-Induced Deformation of a Current-Carrying Wire

3.1 Introduction

Individual yarns and fibrils are able to experience large deformations due to their string-like nature. It is useful to have an analytical solution when verifying numerical methods, but this is difficult with nonlinear problems. A very simple case of a single flexible current carrying wire can be solved. The problem of discussion is illustrated in Figure 3.1.1. The wire is pinned at both ends and an electrical current is set to flow through the wire. An external magnetic field is applied, and the wire will deform in response. Self action will be ignored under the assumption that the magnetic field produced by the current traveling through the wire will be much smaller than the externally applied magnetic field.

First, the well known solution of a helical path of a charge particle traveling through a magnetic field is stated to lay the foundation for the expected behavior of the charge carriers flowing through the conducting wires. Then, the deformation of the wire is solved in a manner similar to the Cateneray solution by treating it as a rigid string (i.e. its length remains constant.) The solution is then refined by treating the wire as an elastic string, with a linear elastic model. The solutions are not unique, and most parameter choices have two valid chiralities and multiple valid circularities. The following chapters will develop a beam based model that are solved numerically and compared to the analytical result.

3.2 Helix Parameterization

A helix can be parameterized by its axis $\mathbf{a}$ that is perpendicular to $\mathbf{e}_r$,

$$\mathbf{x}\left(s\right) = \alpha s\mathbf{a} + r\mathbf{R}\left(\omega s;\mathbf{a}\right)\mathbf{e}_r + \mathbf{d}, \tag{3.2.1}$$

where $\mathbf{R}\left(\omega s;\mathbf{a}\right)$ represents a rotation about the axis $\mathbf{a}$ by angle ωs. A generic rotation matrix is constructed from the axis-angle representation by

$$\mathbf{R}\left(\theta;\mathbf{a}\right) = \cos\theta\mathbf{I} + \sin\theta\left[\mathbf{a}\right]_\times + \left(1-\cos\theta\right)\mathbf{a}\otimes\mathbf{a} \tag{3.2.2}$$

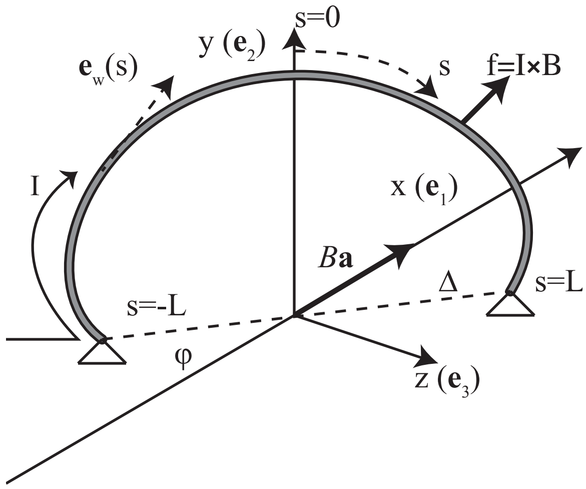


Figure 3.1.1: The pinned wire. The magnetic field is chosen to be oriented along $\mathbf{a} = \mathbf{e}_1$, the pinned boundary conditions are in the xz plane located at $\pm\Delta\mathbf{e}_\Delta$ where the angle between $\mathbf{e}_\Delta$ and $\mathbf{a}$ is equal to ϕ. The wire is of length $2L$, with its parameter s equal to 0 at its center point, which, by symmetry, is located the y axis. The unit vector oriented along the length of the wire is denoted by $\mathbf{e}_w\left(s\right)$.

where $\left[\mathbf{a}\right]_\times$ represents the cross product matrix of $\mathbf{a}$ satisfying $\mathbf{a} \times \mathbf{v} = \left[\mathbf{a}\right]_\times \mathbf{v}$ whose components are

$$\left[\mathbf{a}\right]_\times = \begin{pmatrix} 0 & a_z & -a_y \\ -a_z & 0 & a_x \\ a_y & -a_x & 0 \end{pmatrix}. \tag{3.2.3}$$

The first and second derivatives of the rotation matrix with respect to the angle are

$$\begin{aligned} \frac{d}{ds}\mathbf{R}\left(\omega s;\mathbf{a}\right) &= -\omega \sin\omega s \mathbf{I} + \omega\cos\omega s\left[\mathbf{a}\right]_\times + \omega\sin\omega s \mathbf{a}\otimes\mathbf{a} && (3.2.4) \\ \frac{d^2}{ds^2}\mathbf{R}\left(\omega s;\mathbf{a}\right) &= -\omega^2 \cos\omega s \mathbf{I} - \omega^2\sin\omega s\left[\mathbf{a}\right]_\times + \omega^2\cos\omega s \mathbf{a}\otimes\mathbf{a} && (3.2.5) \end{aligned}$$

Applying the rotation to $\mathbf{e}_2$,

$$\mathbf{R}\left(\omega s;\mathbf{a}\right)\mathbf{e}_r = \cos\omega s\mathbf{e}_r + \sin\omega s\mathbf{a}\times\mathbf{e}_r + \left(1-\cos\omega s\right)\left(\mathbf{a}\cdot\mathbf{e}_r\right)\mathbf{a}. \tag{3.2.6}$$

Since $\mathbf{e}_r$ was picked to be was perpendicular to $\mathbf{a}$, the parameterization and its derivatives are

$$\begin{aligned} \mathbf{x} &= \alpha s\mathbf{a} + r\cos\omega s\mathbf{e}_r + r\sin\omega s\mathbf{a}\times\mathbf{e}_r + \mathbf{d} && (3.2.7) \\ \frac{d\mathbf{x}}{ds} &= \alpha\mathbf{a} - r\omega\sin\omega s\mathbf{e}_r + r\omega\cos\omega s\mathbf{a}\times\mathbf{e}_r && (3.2.8) \\ \frac{d^2\mathbf{x}}{ds^2} &= -r\omega^2\cos\omega s\mathbf{e}_r - r\omega^2\sin\omega s\mathbf{a}\times\mathbf{e}_r. && (3.2.9) \end{aligned}$$

3.3 Helical Particle Trajectory

The ansatz for the wire is motivated by the simple problem of a charged particle in a magnetic field, so that solution is briefly stated here using the notation that will be used in later sections. A particle of mass m and charge q moving with velocity $\mathbf{v}$ observes the Lorentz force in an electromagnetic field, $\mathbf{f} = Q\mathbf{E} + Q\mathbf{v} \times \mathbf{B}$. For a non-relativistic point mass, the equation of motion for the particle is

$$m\dot{\mathbf{v}} = Q\mathbf{v} \times \mathbf{B} \tag{3.3.1}$$

with initial conditions $\mathbf{x}\left(t=0\right) = \mathbf{x}_0$ and $\mathbf{v}\left(t=0\right) = \mathbf{v}_0$. Let the magnetic field be written as $\mathbf{B} = B\mathbf{a}$, where $\mathbf{a}$ is a unit vector that will be the axis—hence the "a"—of the helices. It is typical to pick a coordinate system in which the magnetic field is oriented along one of the principal axes, e.g. $\mathbf{e}_1$, but that will not be the best choice for the next section. Instead, let $\mathbf{a}$ be in some direction and have another unit vector perpendicular to it $\mathbf{e}_r$ that will be in the radial direction of the helix, such that $\mathbf{a} \cdot \mathbf{e}_r = 0$. The velocity can be decomposed along this frame as

$$\mathbf{v} = v_a\mathbf{a} + v_r\mathbf{e}_r + v_\perp\mathbf{a} \times \mathbf{e}_r \tag{3.3.2}$$

The balance of momentum can be written in this frame

$$\frac{d\mathbf{v}}{dt} = -\frac{QB}{m}\mathbf{a} \times \mathbf{v}. \tag{3.3.3}$$

Extracting the component of $\mathbf{v}$ along $\mathbf{a}$ by dotting the above equation by $\mathbf{a}$, we see that ${dv_a}/{dt} = 0$, and therefore the component of the velocity in the same direction of the magnetic field remains unchanged. Since $\mathbf{a} \times \left(\mathbf{a} \times \mathbf{e}_r\right) = -\mathbf{e}_r$,

$$\mathbf{a} \times \mathbf{v} = v_r\mathbf{a} \times \mathbf{e}_r - v_\perp\mathbf{e}_r \tag{3.3.4}$$

and the equations of motion are

$$\begin{aligned} \frac{d}{dt}v_a &= 0 && (3.3.5)\\ \frac{d}{dt}v_2 &= \frac{QB}{m}v_{a2} && (3.3.6)\\ \frac{d}{dt}v_{a2} &= -\frac{QB}{m}v_2. && (3.3.7) \end{aligned}$$

The solution of this problem is a helical trajectory that can be parameterized as in Appendix 3.2. The frequency of the rotation is $\omega = \frac{QB}{m}$ and the radius of the helix is dependent on the component of the initial velocity perpendicular to the field, $r = \frac{v_\perp(0)}{\omega}$. The initial radial unit vector is perpendicular to the initial velocity and magnetic field, $\mathbf{e}_r = \frac{\mathbf{v}_0 \times \mathbf{a}}{||\mathbf{v}_0 \times \mathbf{a}||}$. The solution can then be written as

$$\begin{aligned} \mathbf{x}\left(t\right) &= \left(\mathbf{v}_0 \cdot \mathbf{a}\right) t\mathbf{a} + r\cos\omega t\mathbf{e}_r + r\sin\omega t\mathbf{a} \times \mathbf{e}_r + \mathbf{x}_0 - r\mathbf{e}_r\\ \mathbf{v}\left(t\right) &= \left(\mathbf{v}_0 \cdot \mathbf{a}\right)\mathbf{a} - r\omega\sin\omega t\mathbf{e}_r + r\omega\cos\omega t\mathbf{a} \times \mathbf{e}_r. \end{aligned}$$

3.4 Analytical solution of the shape of a wire in a magnetic field

3.4.1 Balance of linear momentum

Consider a string-like wire with a constant current I in a constant magnetic field $\mathbf{B}$ that is free to flex laterally but, for now, not free to extend elastically. Let a solution be sought that is similar to the Catenary chain solution. The configuration of the wire will be parameterized by s,

$$\mathbf{x}(s) = X(s)\,\mathbf{e}_1 + Y(s)\,\mathbf{e}_2 + Z(s)\,\mathbf{e}_3. \tag{3.4.1}$$

Additionally, the tension in the string, $T(s)$ is unknown.

The solution is constrained to have unit arch length at every point s,

$$\left\| \frac{d\mathbf{x}}{ds} \right\|_2 = 1 \tag{3.4.2}$$

or, in terms of its components,

$$\sqrt{X(s)^2 + Y(s)^2 + Z(s)^2} = 1. \tag{3.4.3}$$

The body force acting on this wire is

$$\mathbf{f}(s) = \rho\mathcal{E} + \mathcal{J} \times \mathbf{B} = \frac{I}{A}\mathbf{e}_w(s) \times \mathbf{B} \tag{3.4.4}$$

where $\mathbf{e}_w(s)$ couples the body force to the orientation of the wire (hence the w) at point s. The orientation of the parameterized curve is

$$\mathbf{e}_w(s) = \frac{{}^{d\mathbf{x}}\!/\!_{ds}}{\left\| {}^{d\mathbf{x}}\!/\!_{ds} \right\|_2} = \frac{d\mathbf{x}}{ds} \tag{3.4.5}$$

which is simplified by the arch length constraint.

Consider now the balance of linear momentum of a segment of the wire from 0 to s, with the tensions at each end of the segment and total force on the body, that can be written as

$$T_0\mathbf{e}_w(0) = \int_0^s I\mathbf{e}_w(s') \times \mathbf{B}ds' + T(s)\,\mathbf{e}_w(s). \tag{3.4.6}$$

The body force integral can be integrated,

$$\begin{aligned} T_0 \left.\frac{d\mathbf{x}}{ds}\right|_0 &= \int_0^s I\frac{d\mathbf{x}}{ds'} \times \mathbf{B}ds' + T(s)\frac{d\mathbf{x}}{ds} && (3.4.7) \\ &= I\left(\int_0^s \frac{d\mathbf{x}}{ds'}ds'\right) \times \mathbf{B} + T(s)\frac{d\mathbf{x}}{ds} && (3.4.8) \\ &= I\left[\mathbf{x}(s')\right]_{s'=0}^{s'=s} \times \mathbf{B} + T(s)\frac{d\mathbf{x}}{ds} && (3.4.9) \end{aligned}$$

to yield an equation for $\mathbf{x}(s)$ and $T(s)$ if T_0 is known and $\mathbf{x}(0)$ is given,

$$T_0 \left. \frac{d\mathbf{x}}{ds} \right|_0 = I \left(\mathbf{x}(s) - \mathbf{x}(0) \right) \times \mathbf{B} + T(s) \frac{d\mathbf{x}}{ds}. \tag{3.4.10}$$

To recover the local differential form of the force balance,

$$\begin{aligned} \mathbf{0} &= \int_0^s I \frac{d\mathbf{x}}{ds'} \times \mathbf{B} ds' + T(s) \frac{d\mathbf{x}}{ds} - T(0) \left. \frac{d\mathbf{x}}{ds} \right|_0 && (3.4.11) \\ &= \int_0^s I \frac{d\mathbf{x}}{ds'} \times \mathbf{B} ds' + \int_0^s \frac{d}{ds'} \left(T(s') \frac{d\mathbf{x}}{ds'} \right) ds' && (3.4.12) \end{aligned}$$

yielding

$$\mathbf{0} = I \frac{d\mathbf{x}}{ds} \times \mathbf{B} + \frac{d}{ds} \left(T \frac{d\mathbf{x}}{ds} \right). \tag{3.4.13}$$

This is the form that will be used in the following sections.

3.4.2 Ansatz

Motivated by the behavior of a single charged particle, an ansatz of a helix is used oriented along the unit vector $\mathbf{a}$ and starting at a vertex $\mathbf{x}(s=0) = r\mathbf{e}_r + \mathbf{d}$, represented by

$$\mathbf{x}(s) = \alpha s \mathbf{a} + r \cos \omega s \mathbf{e}_r + r \sin \omega s \mathbf{a} \times \mathbf{e}_r + \mathbf{d}. \tag{3.4.14}$$

The construction of the parameterization is included in Section 3.2. The $s = 0$ point is placed at the center point of the wire so that its end points are at $s = -L$ and $s = L$. By the symmetry of the problem, $\mathbf{e}_r$ must be perpendicular to both $\mathbf{a}$ and $\mathbf{e}_\Delta$. The choice of bases is arbitrary, but the choice used in plotting the result is diagrammed in Figure 3.1.1: the magnetic field is chosen to be oriented along the x-axis, so that $\mathbf{a} = \mathbf{e}_1$, and $\mathbf{e}_\Delta$ is confined to the xz plane. Therefore, $\mathbf{e}_r$ is oriented along the y-axis, $\mathbf{e}_2$.

The differential form of the force balance relation is

$$\mathbf{0} = \frac{d}{ds} \left(T \frac{d\mathbf{x}}{ds} \right) + \mathbf{f} = \frac{dT}{ds} \frac{d\mathbf{x}}{ds} + T \frac{d^2\mathbf{x}}{ds} + I \frac{d\mathbf{x}}{ds} \times \mathbf{B}. \tag{3.4.15}$$

The helix axis $\mathbf{a}$ is aligned with the magnetic field, so that $\mathbf{B} = B\mathbf{a}$. Plugging into the force balance,

$$\begin{aligned} \mathbf{0} &= \frac{dT}{ds} \left(\alpha \mathbf{a} - r\omega \sin \omega s \mathbf{e}_2 + r\omega \cos \omega s \mathbf{a} \times \mathbf{e}_2 \right) && (3.4.16) \\ &\quad + T \left(-r\omega^2 \cos \omega s \mathbf{e}_2 - r\omega^2 \sin \omega s \mathbf{a} \times \mathbf{e}_2 \right) && (3.4.17) \\ &\quad + IB \left(\alpha \mathbf{a} - r\omega \sin \omega s \mathbf{e}_2 + r\omega \cos \omega s \mathbf{a} \times \mathbf{e}_2 \right) \times \mathbf{a}. && (3.4.18) \end{aligned}$$

The vectors $\mathbf{a}$ and $\mathbf{e}_2$ are chosen to be both unit and orthonormal, so that $(\mathbf{a} \times \mathbf{e}_2) \times \mathbf{a} = \mathbf{e}_2$. Also note that $\mathbf{a} \times \mathbf{a} = 0$ and $\mathbf{e}_2 \times \mathbf{a} = -\mathbf{a} \times \mathbf{e}_2$. Using these relations, the equation can be deconstructed into orthonormal vector components

$$\mathbf{0} = \frac{dT}{ds}\alpha\mathbf{a} + \tag{3.4.19}$$

$$\left(-\frac{dT}{ds}r\omega\sin\omega s\mathbf{e}_2 - Tr\omega^2\cos\omega s + IBr\omega\cos\omega s\right)\mathbf{e}_2 + \tag{3.4.20}$$

$$\left(\frac{dT}{ds}r\omega\cos\omega s - Tr\omega^2\sin\omega s + IBr\omega\sin\omega s\right)\mathbf{a}\times\mathbf{e}_2. \tag{3.4.21}$$

The ansatz satisfies the differential equation with a uniform tension, $\frac{dT}{ds} = 0$, of magnitude

$$T = \frac{IB}{\omega}. \tag{3.4.22}$$

3.4.3 Geometric Boundary Conditions

The wire is pinned at $s = -L$ and $s = L$ at points $\mathbf{x}(L) = \Delta\mathbf{e}_\Delta$ and $\mathbf{x}(-L) = -\Delta\mathbf{e}_\Delta$. The orientation of the helix was chosen such that $\mathbf{e}_2{\cdot}\mathbf{e}_\Delta = 0$. The angle between the magnetic field orientation $\mathbf{a}$ and the supports is ϕ so that $\mathbf{a}\cdot\mathbf{e}_\Delta = \cos\phi$ and $\mathbf{e}_\Delta\times\mathbf{a} = \sin\phi\mathbf{e}_2$.

$$\Delta\mathbf{e}_\Delta = \alpha L\mathbf{a} + r\cos\omega L\mathbf{e}_2 + r\sin\omega L\mathbf{a}\times\mathbf{e}_2 + d\mathbf{e}_2. \tag{3.4.23}$$

The variables will be extracted by dotting against the orthonormal basis formed by $\mathbf{a}$, $\mathbf{e}_2$, and $\mathbf{a}\times\mathbf{e}_2$. Starting with $\mathbf{a}$,

$$\Delta\mathbf{e}_\Delta\cdot\mathbf{a} = (\alpha L\mathbf{a} + r\cos\omega L\mathbf{e}_2 + r\sin\omega L\mathbf{a}\times\mathbf{e}_2 + d\mathbf{e}_2)\cdot\mathbf{a} = \alpha L \tag{3.4.24}$$

resolves α to

$$\alpha = \frac{\Delta\cos\phi}{L}. \tag{3.4.25}$$

Recalling the arc-length constraint $\alpha^2 + r^2\omega^2 = 1$,

$$r = \sqrt{\frac{1-(\Delta\cos\phi)^2}{\omega^2L^2}}. \tag{3.4.26}$$

Proceeding to $\mathbf{a}\times\mathbf{e}_2$,

$$\Delta\mathbf{e}_\Delta\cdot(\mathbf{a}\times\mathbf{e}_2) = r\sin\omega L \tag{3.4.27}$$

and cycling the scalar triple product yields the equation for the quantity ωL

$$\frac{\Delta\sin\phi}{\sqrt{1-(\Delta\cos\phi)^2}} = \frac{\sin\omega L}{\sqrt{(\omega L)^2}}. \tag{3.4.28}$$

This equation must be solved numerically, but restricting the search to $\omega > 0$ simplifies the equation to the familiar equation $\frac{\sin x}{x}$ for $x = \omega L$. The final parameter can be determined by dotting with $\mathbf{e}_2$,

$$d = -r\cos\omega L. \tag{3.4.29}$$

3.4.4 Nondimensionalization

The length values can be nondimensionalized by dividing by L, effectively setting $L = 1$: $s' = \frac{s}{L}$, $p' = \frac{p}{L}$, $\mathbf{x}' = \frac{\mathbf{x}}{L}$. The ansatz has the form

$$\mathbf{x}'\left(p'\right) = \alpha p' \mathbf{a} + r\mathbf{R}\left(\omega' p'; \mathbf{a}\right)\mathbf{e}_2 + \mathbf{d}' \tag{3.4.30}$$

where the parameters are scaled by $\mathbf{d}' = \frac{\mathbf{d}}{L}$ and $\omega' = \omega L$. The clamped boundary condition distance is scaled as

$$\Delta' = \frac{\Delta}{L} \tag{3.4.31}$$

The forces can be nondimensionalized by dividing by EA to obtain a characteristic force-per-unit-length-per-unit-stiffness of

$$k = \frac{IB}{EA}. \tag{3.4.32}$$

The problem can then be specified by only three values: the ratio of the clamp distance to the length of the wire, Δ'; the angle of the magnetic field to the supports, ϕ; and the ratio of the applied force-per-length to the stiffness of the wire, k.

3.4.5 Elastic Wire

Now, instead of a chain, consider a linear elastic wire whose tension is linearly related to its strain at a point, such that

$$T\left(s\right) = EA\varepsilon\left(s\right) = EA\left(U\left(s\right) - 1\right). \tag{3.4.33}$$

Two coordinate frames are defined running along the length of the wire: p in the unstressed reference configuration and s in the stretched configuration. The coordinates are related by

$$\frac{ds}{dp} = 1 + \varepsilon = U. \tag{3.4.34}$$

The constraint $\left\|{}^{d\mathbf{x}}/{}_{ds}\right\|_2 = 1$ remains, but now the curve is parameterized along p, yielding the relation

$$\left\|\frac{d\mathbf{x}}{dp}\right\|_2 = U\left(p\right). \tag{3.4.35}$$

The force balance equation is in the real length parameter, so it is pushed back into the reference configuration to form the solution there. Plugging the constitutive law for the tension into the force balance equation,

$$\begin{aligned}
\mathbf{0} &= \frac{dT}{ds}\frac{d\mathbf{x}}{ds} + T\frac{d^2\mathbf{x}}{ds^2} + I\frac{d\mathbf{x}}{ds} \times \mathbf{B} && (3.4.36)\\
&= EA\frac{dU}{ds}\frac{d\mathbf{x}}{ds} + EA\left(U - 1\right)\frac{d^2\mathbf{x}}{ds^2} + I\frac{d\mathbf{x}}{ds} \times \mathbf{B} && (3.4.37)\\
&= EA\frac{1}{U^2}\frac{dU}{dp}\frac{d\mathbf{x}}{dp} + EA\frac{\left(U - 1\right)}{U^2}\frac{d^2\mathbf{x}}{dp^2} + I\frac{1}{U}\frac{d\mathbf{x}}{dp} \times \mathbf{B}. && (3.4.38)
\end{aligned}$$

Multiplying by U^2,

$$\mathbf{0} = EA\frac{dU}{dp}\frac{d\mathbf{x}}{dp} + EA\left(U-1\right)\frac{d^2\mathbf{x}}{dp^2} + IBU\frac{d\mathbf{x}}{dp} \times \mathbf{a}. \tag{3.4.39}$$

Using the above defined nondimensional variables, the above equation is

$$\mathbf{0} = \frac{dU}{dp}\frac{d\mathbf{x}}{dp} + \left(U-1\right)\frac{d^2\mathbf{x}}{dp^2} + kU\frac{d\mathbf{x}}{dp} \times \mathbf{a}. \tag{3.4.40}$$

As in the rigid string case, substituting and collecting the components yields

$$\begin{aligned} \mathbf{0} &= \alpha\frac{dU}{dp}\mathbf{a} + && (3.4.41)\\ &\left(-\frac{dU}{dp}r\omega\sin\omega p - \left(U-1\right)r\omega^2\cos\omega p + kUr\omega\cos\omega p\right)\mathbf{e}_2 && (3.4.42)\\ &\left(\frac{dU}{dp}r\omega\cos\omega p - \left(U-1\right)r\omega^2\sin\omega p + kUr\omega\sin\omega p\right)\mathbf{a}\times\mathbf{e}_2 && (3.4.43) \end{aligned}$$

It is determined that the stretch is constant, ${}^{dU}/_{dp} = 0$, and equal to

$$U = \frac{\omega}{\omega - k}. \tag{3.4.44}$$

3.4.6 Geometric boundary conditions for the elastic wire

Parametrized with p now, the position of the cure has the same form

$$\mathbf{x}\left(p\right) = \alpha p\mathbf{a} + r\mathbf{R}\left(\omega p;\mathbf{a}\right)\mathbf{e}_2 + \mathbf{d} \tag{3.4.45}$$

except that the arclength constrain is related to the stretch, yielding $\alpha^2 + r^2\omega^2 = U^2$. Following the same procedure as before, where α is unchanged,

$$r = \sqrt{\frac{U^2-\alpha^2}{\omega^2}} = \sqrt{\frac{1}{\left(\omega-k\right)^2} - \frac{\left(\Delta\cos\phi\right)^2}{\omega^2}}. \tag{3.4.46}$$

Note that in the limit as $k \to 0$, or $EA \gg IB$, r approaches the value for the rigid chain. The last boundary condition constraint of $\Delta\sin\phi = r\sin\omega$ yields the equation

$$\Delta\sin\phi = \sin\omega\sqrt{\frac{1}{\left(\omega-k\right)^2} - \frac{\left(\Delta\cos\phi\right)^2}{\omega^2}} \tag{3.4.47}$$

which must be solved to obtain ω.

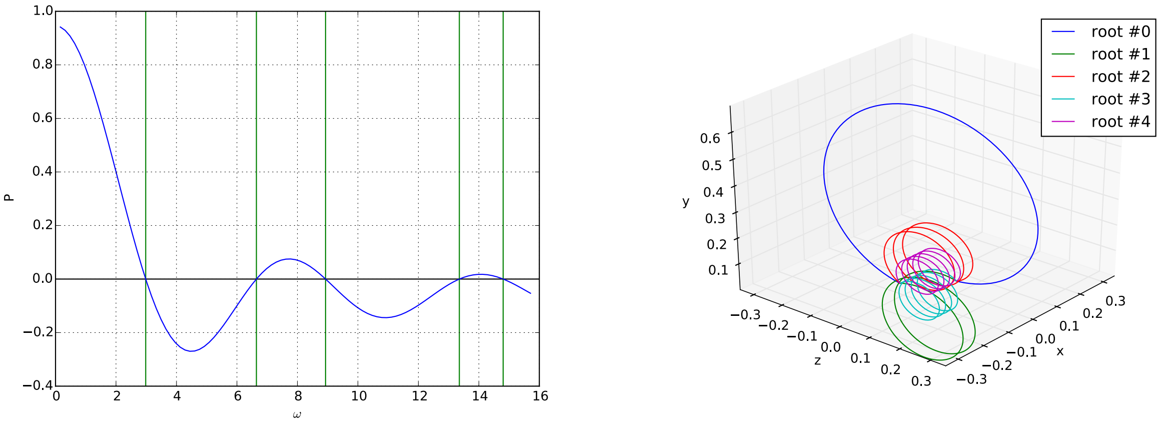


Figure 3.5.1: The multiple solutions for the deformation for $\Delta = 0.075$, $\phi = \frac{\pi}{4}$, and $k = 0$. The constraint equation for ω is shown on the left, with the roots marked with vertical lines. The roots are numbered sequentially increasing with the magnitude of ω. Only the positive roots of ω are shown, with the positive choice of r.

3.4.7 Strain Energy

The total strain energy in the wire can be calculated by integrating over the nondimensionalized length of the wire,

$$W = \int_{\Omega} \varepsilon : \sigma d\Omega = \int_{-1}^{1} EAU^2 ds. \tag{3.4.48}$$

Since the strain is a constant, the total strain energy is

$$W = \underbrace{2A}_{=Vol} E \frac{\omega^2}{(\omega - k)^2}. \tag{3.4.49}$$

3.5 Solution Characterization

3.5.1 Number of roots

The constraint for ω in Equation 3.4.47 has multiple roots. Ordering the roots starting at one, the number of the root corresponds to the circularity of the wire. Only a finite number of roots exists, as after a certain amount of turns the wire is not long enough to reach the clamps. The exception is the case when $\Delta = 0$ and the two pins are coincident, allowing the wire to turn infinitely many times.

3.5.2 Negative roots

Negative roots for ω in the constraint equation exist. These are valid solutions to the problem, but recalling the form for the tension in the rigid case, $T = \frac{IB}{\omega}$, these roots

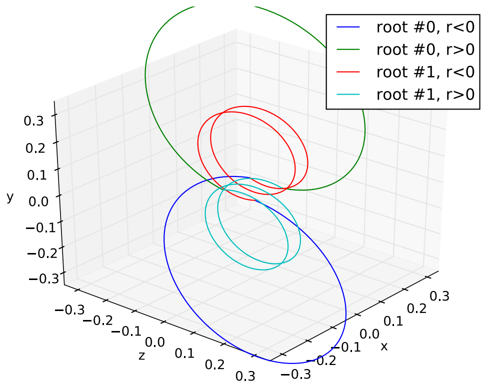


Figure 3.5.2: Deformation for both signs of r using parameters $\Delta = 0.075$, $\phi = \frac{\pi}{4}$, and $k = 0$. First two roots of ω are shown.

correspond to compressive modes. That is, the negative roots correspond to the flexible rod in compression, where the applied force wants the helix to collapse on itself since it is "inside out" from the charge carriers' perspective. Considering the elastic rod, the tension

$$T = EA\left(\frac{\omega}{\omega - k} - 1\right)$$

will be compressive, $T < 0$, when $\omega < k$. This agrees with the rigid case which corresponds to the limit $k \to 0$.

3.5.3 Sign of r

Because r is calculated by a square root, both positive and negative values are possible for each root. These two cases are plotted in Figure 3.5.2. Both cases have the same handedness and correspond to tension, but are on opposite sides of the plane. Considering a situation under which the wire is undergoing dynamic relaxation initially running straight between the two pins, the forces will be pointing in the positive y direction initially, leading it to a solution that exists in the positive y plane.

3.5.4 Variation of parameters.

The effects of varying the angle between the magnetic field and the pins are shown in 3.5.3. The effect of decreasing the stiffness of the wire is shown in Figure 3.5.4.

3.6 Conclusion

The solution derived in this chapter can be summarized as follows:

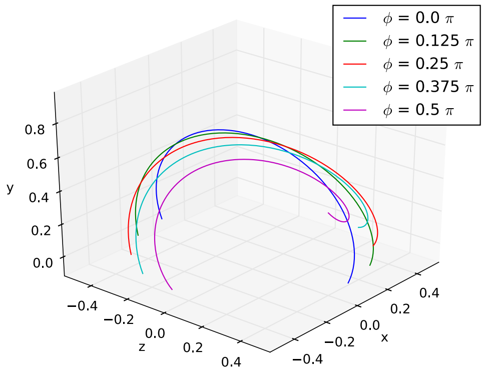


Figure 3.5.3: Deformations for varying the angle of the magnetic field, ϕ, using $\Delta = 0.5$ and $k = 0$. Only one root of ω and r are shown for each case.

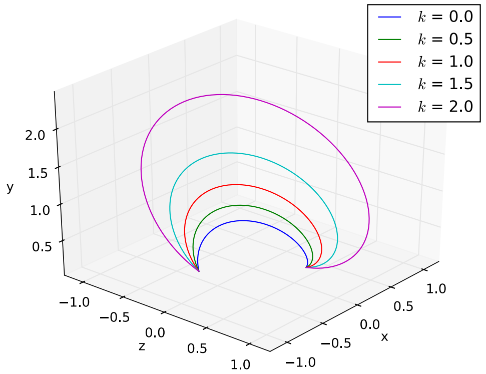


Figure 3.5.4: Deformations for decreasing the stiffness of the wire, k, using $\Delta = 0.5$ and $\phi = \frac{\pi}{4}$. Only one root of ω and r are shown for each case.

1. Define the nondimensional parameters:

$$\Delta' = \frac{\Delta}{L}, \mathbf{x}' = \frac{\mathbf{x}}{L}$$

$$k = \frac{IB}{EA}$$

2. Solve the following equation for ω:

$$\Delta \sin\phi = \sin\omega \sqrt{\frac{1}{(\omega - k)^2} - \frac{(\Delta\cos\phi)^2}{\omega^2}}$$

3. Compute

$$\begin{aligned} \alpha &= \Delta\cos\phi \\ r &= \sqrt{\frac{1}{(\omega - k)^2} - \frac{(\Delta\cos\phi)^2}{\omega^2}} \\ d &= -r\cos\omega \end{aligned}$$

4. The solution with $p \in [-1, 1]$ is

$$\mathbf{x}(p) = \alpha p\mathbf{a} + r\cos\omega s\mathbf{e}_r + r\sin\omega s\mathbf{a} \times \mathbf{e}_r + d\mathbf{e}_r$$

Multiple values of ω and both signs of r can be valid for a given set of parameters. The successive roots of ω correspond to increasing circularities, with a maximum value constrained by the length of the string. The two signs of r correspond to tension or compression. The existence of multiple solutions causes instabilities that will be a challenge in static analysis of the numerical method.

Chapter 4

A Fibril Assembly Model of Textile Microstructure

4.1 Introduction

Each fibril is described using a director-based beam model with an assumed kinematic displacement field. The solutions are constructed with a one dimensional finite element basis along the axis of the fibrils and an ansatz along the cross section. This type of beam theory is explained in Rubin [103]. Additionally, thermal and electromagnetic fields are incorporated into the beam formulation. Using a beam formulation has the following advantages:

1. arbitrary cross sections can be exactly represented geometrically,
2. the discretization cost is greatly reduced, and
3. the electromagnetic problem can be simplified.

The cross section integration is performed using Gaussian quadrature, which eases the implementation of complex multiphysics couplings. Contact between beams is also considered, with jump conditions on the thermal and electromagnetic problems treated.

4.2 Formulation

4.2.1 Finite Deformation Kinematic beam model

Let ξ_3 denote the parameter of the axis running along the fibril in a reference configuration. To simplify the solution, the fibrils are assumed to have linear displacement fields along their cross section, directions ξ_1 and ξ_2. Let $\mathbf{r}\left(\xi_3\right)$ denote the position of the centroid of a given cross section of the fiber, and $\mathbf{g}_1\left(\xi_3\right)$ and $\mathbf{g}_2\left(\xi_3\right)$ denote directors of the cross section in the directions ξ_1 and ξ_2. These fields are assembled into an assumed solution for the position field

$$\mathbf{x}\left(\xi_1, \xi_2, \xi_3, t\right) = \mathbf{r}\left(\xi_3, t\right) + \xi_1 \mathbf{g}_1\left(\xi_3, t\right) + \xi_2 \mathbf{g}_2\left(\xi_3, t\right).$$

The director fields are illustrated in Figure 4.2.1. The coordinate system ξ was defined for convenient description of the geometry of the long and thin fiber to allow this simplified deformation description. The natural (or initial, or unstrained) configuration, $\mathbf{X}_0$, may differ

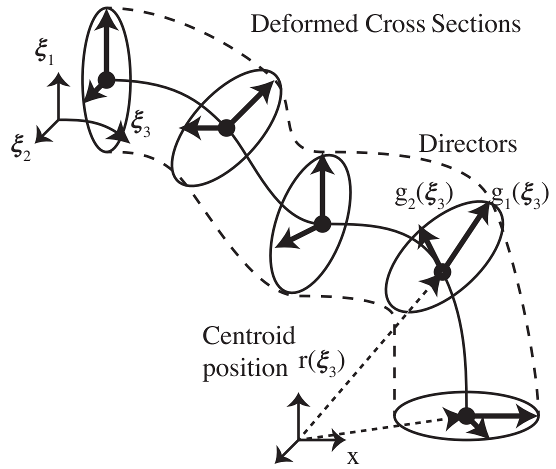


Figure 4.2.1: Director based beam formulation

from this reference configuration; i.e., a fiber may be naturally bent, but it is still easier to described its deformation using the curvilinear system ξ. The configurations were illustrated in Figure 2.3.1.

The ansatz can also be formulated in terms of the displacement, introducing new fields $\mathbf{q}$, $\mathbf{h}_1$, and $\mathbf{h}_2$ satisfying

$$\begin{aligned}\mathbf{r}\left(\xi_3,t\right) &= \mathbf{r}_0\left(\xi_3\right)+\mathbf{q}\left(\xi_3,t\right)\\ \mathbf{g}_1\left(\xi_3,t\right) &= \mathbf{g}_{01}\left(\xi_3\right)+\mathbf{h}_1\left(\xi_3,t\right)\\ \mathbf{g}_2\left(\xi_3,t\right) &= \mathbf{g}_{02}\left(\xi_3\right)+\mathbf{h}_2\left(\xi_3,t\right)\end{aligned}$$

that form the displacement field

$$\mathbf{u}\left(\xi_1,\xi_2,\xi_3,t\right)=\mathbf{q}\left(\xi_3,t\right)+\xi_1\mathbf{h}_1\left(\xi_3,t\right)+\xi_2\mathbf{h}_2\left(\xi_3,t\right).$$

The subscript 0 is used to denote fields evaluated in their reference state. Since the curvilinear coordinate system of the reference configuration is fixed, the velocity is related to the director and displacement fields by

$$\begin{aligned}\mathbf{v} &= \frac{d}{dt}\mathbf{x}\left(\xi_1,\xi_2,\xi_3,t\right)\\ &= \frac{d\mathbf{r}}{dt}\left(\xi_3,t\right)+\xi_1\frac{d\mathbf{g}_1}{dt}\left(\xi_3,t\right)+\xi_2\frac{d\mathbf{g}_2}{dt}\left(\xi_3,t\right)\\ &= \frac{d\mathbf{q}}{dt}\left(\xi_3,t\right)+\xi_1\frac{d\mathbf{h}_1}{dt}\left(\xi_3,t\right)+\xi_2\frac{d\mathbf{h}_2}{dt}\left(\xi_3,t\right).\end{aligned}$$

When discussing the discretization and implementation later, the rates of the deformation descriptors will be prefixed with a $\mathbf{v}$ instead of $\frac{d}{dt}$, and the velocity will be written as

$$\begin{aligned}\mathbf{v}\left(\xi_1,\xi_2,\xi_3,t\right) &= \mathbf{vr}\left(\xi_3,t\right)+\xi_1\mathbf{vg}_1\left(\xi_3,t\right)+\xi_2\mathbf{vg}_2\left(\xi_3,t\right)\\ &= \mathbf{vq}\left(\xi_3,t\right)+\xi_1\mathbf{vh}_1\left(\xi_3,t\right)+\xi_2\mathbf{vh}_2\left(\xi_3,t\right).\end{aligned}$$

In this representation, the deformation at any given time is fully described by six vector fields that vary along ξ_3, namely by the positions $\mathbf{r}, \mathbf{g}_1, \mathbf{g}_2, \mathbf{vr}, \mathbf{vg}_1$, and $\mathbf{vg}_2$ or by the displacements $\mathbf{q}, \mathbf{h}_1, \mathbf{h}_2, \mathbf{vq}, \mathbf{vh}_1$ and $\mathbf{vh}_2$.

The deformation gradient for this field can be calculated by using the chain rule,

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \frac{\partial \mathbf{x}}{\partial \xi}\frac{\partial \xi}{\partial \mathbf{X}_0} = \frac{\partial \mathbf{x}}{\partial \xi}\left(\frac{\partial \mathbf{X}_0}{\partial \xi}\right)^{-1}$$

where $\mathbf{X}_0$ is the mapping from the ideal configuration ξ to the reference configuration $\mathbf{X}$. The gradient with respect to the ideal configuration is

$$\frac{\partial \mathbf{x}}{\partial \xi} = \left(\frac{\partial \mathbf{r}}{\partial \xi_3} + \xi_1 \frac{\partial \mathbf{g}_1}{\partial \xi_3} + \xi_2 \frac{\partial \mathbf{g}_2}{\partial \xi_3}\right) \otimes \mathbf{E}_3 + \mathbf{g}_1 \otimes \mathbf{E}_1 + \mathbf{g}_2 \otimes \mathbf{E}_2.$$

If the unstrained configuration is assumed to be identical to the reference configuration, $\mathbf{X}_0 = \xi$ as is the case for an initially straight fiber, then $\frac{\partial \mathbf{X}_0}{\partial \xi} = \mathbf{1}$, and the deformation gradient is $\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \xi}$. The spatial velocity gradient is required for some rate dependent models. Evaluating this quantity similarly to the deformation gradient,

$$\mathbf{L} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}} = \frac{\partial \mathbf{v}}{\partial \mathbf{X}}\left(\frac{\partial \mathbf{x}}{\partial \mathbf{X}}\right)^{-1} = \frac{\partial \mathbf{v}}{\partial \xi}\left(\frac{\partial \mathbf{x}}{\partial \xi}\right)^{-1}.$$

Knowledge the reference configuration is not necessarily required for the calculation, but the current configuration $\mathbf{x}$ is required. The gradient of the velocity with respect to the reference configuration is

$$\frac{\partial \mathbf{v}}{\partial \xi} = \left(\frac{\partial \mathbf{vr}}{\partial \xi_3} + \xi_1 \frac{\partial \mathbf{vg}_1}{\partial \xi_3} + \xi_2 \frac{\partial \mathbf{vg}_2}{\partial \xi_3}\right) \otimes \mathbf{E}_3 + \mathbf{vg}_1 \otimes \mathbf{E}_1 + \mathbf{vg}_2 \otimes \mathbf{E}_2.$$

Computing the spatial velocity gradient $\mathbf{L}$ at a particular time requires inversion of $\frac{\partial \mathbf{x}}{\partial \xi}$.

An important note is that when the the ideal coordinates and reference coordinates coincide (or are at least scaled the same), the directors are of unit length in the reference configuration, e.g. $|\mathbf{g}_{01}| = |\mathbf{g}_{02}| = 1$. If in the present configuration, the cross section is simply rotated, the deformed directors will also be of unit length.

The unit normal to the *cross section* can be obtained by

$$\mathbf{n}_C\left(\xi_3,\right) = \frac{\mathbf{g}_1 \times \mathbf{g}_2}{||\mathbf{g}_1 \times \mathbf{g}_2||_2}.$$

The cross section and centroid are not required to be perpendicular in the director-based theory. The unit vector along the centroid will be useful later—the electric current will be assumed to run along the centroid, not normal the cross section—so it is defined here in terms of the centroid position

$$\mathbf{e}_y\left(\xi_3\right) = \frac{{}^{d\mathbf{r}}\!/_{d\xi_3}}{||{}^{d\mathbf{r}}\!/_{d\xi_3}||_2} = \frac{\mathbf{E}_3 + {}^{d\mathbf{q}}\!/_{d\xi_3}}{||\mathbf{E}_3 + {}^{d\mathbf{q}}\!/_{d\xi_3}||_2}$$

where $\mathbf{E}_3 = \frac{d\mathbf{r}_0}{d\xi_3}$ is the unit vector along the centroid in the ideal configuration.

The assumed displacement field can be plugged directly into the constitutive relations from balance of linear momentum,

$$\rho \dot{\mathbf{v}} \quad = \quad \nabla_x \cdot \mathbf{T}\left(\mathbf{u}, ...\right) + \rho \mathbf{f}$$

to create a vector equation with three unknown time-varying vector fields instead of only one,

$$\rho \frac{d}{dt}\left(\mathbf{vq} + \xi_1 \mathbf{vh}_1 + \xi_2 \mathbf{vh}_2\right) = \nabla_x \cdot \mathbf{T}\left(\mathbf{q}, \mathbf{h}_1, \mathbf{h}_2, ...\right) + \rho \mathbf{f}$$

not counting the additional unknown fields, T and V.

Remark: Locking

The director formulation as described exhibits bending-mode locking whenever the Poisson ratio is not zero. This locking mode is observed in standard linear nodal-elements [2]. Increasing the order of the interpolation does not alleviate the problem, as the displacement field remains linear with respect to the cross section by the ansatz inherent to the formulation. Various techniques for correcting this problem can be employed. This behavior with respect to this beam formulation is discussed in [104]. In [38], the strain components yielding the locking are treated as constants along the cross section. Since this aspect of the beam formulation itself is not the main topic of this work, the problem is side-stepped by only considering the case $\nu = 0$ in all of the results presented. Both the formulation and implementation are flexible enough to accommodate a correction to the locking issue.

4.2.2 The Restriction of Electromagnetic Problem to the Beam

Without further assumptions, it would be necessary to mesh the space around the textile, even if it were a vacuum, and calculate the electromagnetic fields there as well. For this problem, it is assumed that the magnetic field is dominated by an externally applied source, such that the contributions by currents in the fabric are negligible. Only the electric field and current inside of the textile will be calculated during the simulation. The electric field outside of the material varies as a consequence but is not considered. The only appreciable boundary conditions on the electric field will be connections between the fibrils and some external electric circuit. The contributions of external electric fields and of polarization on the cross sections of the fibrils are neglected, allowing the interstitial and far-field space to not be discretized.

Continuing with the beam-like fibrils assumptions, the base assumption to the electromagnetic discretization will be that the conduction current density will be uniform across the cross section and oriented along the axis of the beam,

$$\boldsymbol{\mathcal{J}} = \mathcal{J}\left(\xi_3\right) \mathbf{e}_y\left(\xi_3\right).$$

(Additional currents will appear at contacts, but those will be restricted to exist only at the surfaces and their contribution to the current density field inside of the fibril is neglected.) The notations used will now collide slightly with respect to the electric field and the reference coordinate basis: $\mathbf{FE}_3 \parallel \mathbf{e}_y$, $\mathbf{FE}_1$, and $\mathbf{FE}_2$ are the basis vectors of the reference coordinate system pushed into the current configuration. They will be used now as vectors in the spatial coordinate frame, i.e. with their components expressed with respect to x. Using Ohm's law, $\boldsymbol{\mathcal{J}} = \sigma \boldsymbol{\mathcal{E}}$, the electromotive intensity must also be oriented along the axis of the fibril, i.e. $\boldsymbol{\mathcal{E}} \cdot \mathbf{F}\mathbf{E}_1 = \boldsymbol{\mathcal{E}} \cdot \mathbf{F}\mathbf{E}_2 = 0$.

The electric (voltage) potential will be used as the discretized field to be solved. The electric potential is not invariant, but the invariant quantity $V' = V - \mathbf{v} \cdot \mathbf{A}$ is unwieldy for this situation due to its dependence of the magnetic vector potential. Let V be the *Lab Frame* electric potential, which satisfies

$$\mathbf{E} = -\nabla_x V$$

in the absence of a time varying vector potential, i.e. $\frac{\partial \mathbf{A}}{\partial t}$. The gradient of the potential can be broken up into components by

$$\nabla_\xi V = \frac{\partial V}{\partial \xi_3}\mathbf{E}_3 + \frac{\partial V}{\partial \xi_1}\mathbf{E}_1 + \frac{\partial V}{\partial \xi_2}\mathbf{E}_2.$$

Using the above assumption, the lab frame electric field can be broken up into the curvilinear coordinates of the beam as $(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \mathbf{F}\mathbf{E}_1 = (\mathbf{E} + \mathbf{v} \times \mathbf{B}) \cdot \mathbf{F}\mathbf{E}_2 = 0$. This yields the following restriction for the electric potential:

$$\begin{aligned} \mathbf{F}^T \nabla_x V \cdot \mathbf{E}_1 &= \mathbf{F}^T (\mathbf{v} \times \mathbf{B}) \cdot \mathbf{E}_1 \\ \mathbf{F}^T \nabla_x V \cdot \mathbf{E}_2 &= \mathbf{F}^T (\mathbf{v} \times \mathbf{B}) \cdot \mathbf{E}_2. \end{aligned}$$

Even though the potential is being constructed in the lab frame, the condition that current must flow along the axis of the beam yields a trivial condition on the gradient of the voltage across the cross section. These components could be solved for by plugging the the ansatz for $\mathbf{v}$, but they do not have any effect on the current, so they are uninteresting. Returning to the potential equation and breaking the divergence in the components along the ideal coordinates,

$$0 = \nabla_\xi \cdot \epsilon_{r\xi} \nabla_\xi V - \nabla_\xi \cdot \epsilon_{r\xi} J \left(\frac{\partial \mathbf{x}}{\partial \boldsymbol{\xi}}\right)^{-1} \mathbf{v} \times \mathbf{B}.$$

Only the ξ_3 direction is not trivially satisfied, so

$$\frac{\partial}{\partial X_3} \epsilon_{r0} \frac{\partial V}{\partial X_3} = \frac{\partial}{\partial X_3} \epsilon_{r0} \left(\mathbf{v} \times \mathbf{B} \cdot \mathbf{F}\mathbf{E}_3\right). \tag{4.2.1}$$

Thus, the beams can be assumed to have a constant voltage along their cross section, so that it is only a function of the axis coordinate,

$$V = V(\xi_3).$$

Since the only non-zero component of $\boldsymbol{\mathcal{E}}$ is along the axis coordinate, the conduction current density in the laboratory frame can be determined to depend on the laboratory frame voltage, the deformation gradient, and the material velocity by

$$\boldsymbol{\mathcal{J}} = \frac{1}{J}\mathbf{F}\sigma_0 \left(-\frac{\partial V}{\partial \xi_3} + \mathbf{F}^T \mathbf{v} \times \mathbf{B} \cdot \mathbf{E}_3\right)\mathbf{E}_3.$$

The electric charge density is calculated as a side effect to solving Equation 4.2.1 by way of the the quasistatic solution of Equation 2.8.1. It can be post-processed using this equation, and would be required to calculate the force density to obtain the equation $q\boldsymbol{\mathcal{E}} = -\left(\nabla_x \cdot \epsilon_0 \mathbf{v} \times \mathbf{B}\right)\boldsymbol{\mathcal{E}}$. However, even though the charge density is implied by the formulation, the post-processing step is extra computational work. The magnitude of the force density $q\boldsymbol{\mathcal{E}} = q\mathbf{E} + q\mathbf{v} \times \mathbf{B}$ is much smaller than the magnitude of the term $\boldsymbol{\mathcal{J}} \times \mathbf{B}$ and is neglected. Consider a $1mm$ long yarn segment with a $0.1mm$ cross sectional radius moving perpendicular to a magnetic field of $1T$ with a velocity of $100 {}^{m}\!/\!{}_{s}$ and $0 {}^{m}\!/\!{}_{s}$ at either end. A back-of-the-envelop calculation shows that the force density magnitude $\left(\epsilon_0 \frac{\Delta v}{L} B\right)(vB) = 8.85 \times 10^{-12} {}^{F}\!/\!{}_{m} \times 100 {}^{m}\!/\!{}_{s} \times 1T/10^{-3}m$ is approximately $8.85 \times 10^{-5} {}^{N}\!/\!{}_{m^3}$. For comparison, the force density if the yarn was carrying $1A$ of current $IB/A = 1A \times 1T/\left(\pi\left(0.1mm\right)^2\right)$ is approximately $3.1 \times 10^{7} {}^{N}\!/\!{}_{m^3}$. Thus, the force density for the beam is only calculated by

$$\mathbf{f} = \boldsymbol{\mathcal{J}} \times \mathbf{B}.$$

4.2.3 Balance of Energy

The beams are assumed to have a constant temperature across their cross section, so that the temperature is only a function of the axis coordinate,

$$T = T\left(\xi_3\right).$$

This representation can be plugged into the balance of energy laws directly. Using the above relations for the electromagnetic formulation, the Joule heating term can be calculated by

$$\boldsymbol{\mathcal{J}} \cdot \boldsymbol{\mathcal{E}} = \frac{1}{J}\sigma_0 \left(-\frac{\partial V}{\partial \xi_3} + \mathbf{F}^T \mathbf{v} \times \mathbf{B} \cdot \mathbf{E}_3\right)^2.$$

4.3 Contact Treatment

4.3.1 Surface Mapping

To integrate the contact problem into a general finite element framework, contacts can be viewed a surface of discontinuity. The treatment of the constitutive responses of the surfaces can be described using jump conditions on the change in fields across the two material bodies. One body can be referred to as the + surface and the other as the − surface, where the normal vector $\mathbf{n}$ will be defined to point towards the + surface, as illustrated in Figure 4.3.1. Each body has two separate problems and two separate reference configurations,

$$\begin{aligned} \mathbf{x}^+ &= \chi^+\left(\mathbf{X}^+\right) \\ \mathbf{x}^- &= \chi^-\left(\mathbf{X}^-\right) \end{aligned}$$

The surface can be parametrized by two coordinates, u, v, with its own position function

$$\mathbf{x}_C = \mathbf{x}_C\left(u, v\right).$$

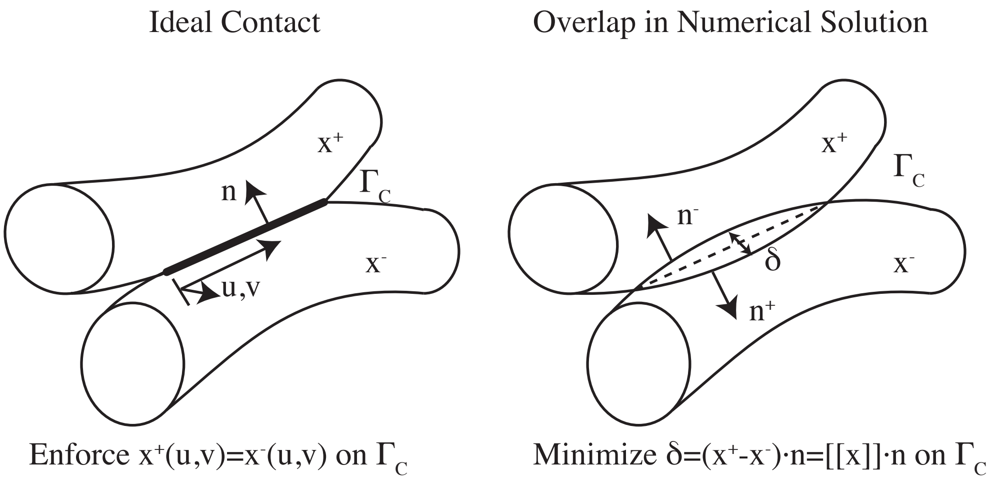

Ideal Contact
Overlap in Numerical Solution
x^+
n
Γ_C
x^-
u,v
Enforce $x^+(u,v)=x^-(u,v)$ on Γ_C
x^+
Γ_C
n^-
δ
x^-
n^+
Minimize $\delta=(x^+-x^-)\cdot n=[[x]]\cdot n$ on Γ_C

Figure 4.3.1: Contact constraint

Ideally, on a contact, the two surfaces will be coincident and have the same spatial coordinates, i.e.

$$\mathbf{x}_C\left(u,v\right) = \mathbf{x}^+\left(u,v\right) = \mathbf{x}^-\left(u,v\right) \text{ on } \Gamma_C.$$

The mapping from the contact surface to each of the reference configurations can be constructed from the inverse of the two deformation mappings,

$$\begin{aligned} \mathbf{X}^+\left(u,v\right) &= {\chi_+}^{-1}\left(\mathbf{x}_C\left(u,v\right)\right) \\ \mathbf{X}^-\left(u,v\right) &= \chi_-^{-1}\left(\mathbf{x}_C\left(u,v\right)\right). \end{aligned}$$

(The body specifiers + and − will move to the subscript when there is an exponent present.)

The condition that the surfaces are coincident will be weakened to enable finding finite element solutions to satisfy it. As illustrated in Figure 4.3.1, the approximate solutions to each body will be allowed to overlap slightly. The contact region Γ_C is the portion on each surface where the two bodies overlap. The mapping between the two surfaces is needed, as they are no longer coincident, and the contact region is now considered to be *two connected surfaces.* No intermediate geometry is defined in this formulation. The region is defined as pairs of points on each surface that are inside of the other body that are the closest to one another. That is, for each point on the boundary of one body and inside the other body, its connected point is the closest point on the other body's surface. The geometry can be defined by and constructed by

$$\Gamma_C = \left\{ \left(\mathbf{x}^+, \mathbf{x}^-\right) | \forall \mathbf{x}^+ \in \left(\partial\Omega^+ \cap \Omega^-\right), \mathbf{x}^- = \arg \min_{\mathbf{x}^- \in (\Omega^+ \cap \partial\Omega^-)} \left|\mathbf{x}^+ - \mathbf{x}^-\right| \right\}.$$

This object directly translates into a table of quadrature points pairs in the finite element implementation; the procedure for its construction is described in Section 4.3.4. The region can still be parametrized by two coordinates u, v, but all that matters is that some labeling scheme for the connected pairs exists. The jump in the spatial coordinate on the contact region is equal to

$$[[\mathbf{x}]] = \mathbf{x}^{+} - \mathbf{x}^{-}.$$

Since the surfaces are not exactly in a conforming contact, the surface normals along the contact for each body may not be in agreement, i.e.$\mathbf{n}^{+} \neq \mathbf{n}^{-}$. Therefore, some approximation must be used, such as the average normal, $\langle \mathbf{n} \rangle = \frac{1}{2}\left(\mathbf{n}^{+} + \mathbf{n}^{-}\right)$. The beam geometry utilized will yield an applicable equation for the normal vector.

4.3.2 Constitutive laws

The contact region represents a surface of discontinuity for all of the fields that exist on the two bodies. Thus, it is needed to specify constitutive responses for the tractions, heat fluxes, and surface currents that are functions of the jumps in the fields,

$$\{\mathbf{t}, q_n, k_n\} = f\left([[\mathbf{x}]], [[T]], [[V]]\right).$$

They may all be dependent on one another; particularly the contact pressure has significant effects on both thermal and electric contact resistances[62]. For simplicity, only linear and independent contact laws are used in this work. For the tractions, only a simple penalty term with a sufficiently large parameter P^* is used to penalized the overlap,

$$\mathbf{t} = P^* \left([[\mathbf{x}]] \cdot \mathbf{n}\right) \mathbf{n}.$$

The two bodies are also placed in thermal contact, so a jump condition is needed to relate the heat flux across the bodies that is dependent on the jump in temperature. Linear thermal contact resistances have the form

$$\mathbf{q} = h_c \, [[T]] \, \mathbf{n}$$

where h_c is the contact heat transfer coefficient. Similarly, for the electromagnetic problem, a contact resistance appears with a surface conductivity σ_c that relates the current across the surface to the potential drop along the surface,

$$\mathbf{k} \;\; = \;\; \sigma_c \, [[V]] \, \mathbf{n}.$$

4.3.3 Beam Geometry

The beam-based theory has a particular geometry representation that needs to be taken into account to handle the contacts. The geometry is illustrated in Figure 4.3.2. Since the finite element meshes are one dimensional along the axis, instead of preventing an overlap of two solid finite element meshes, the penalty must be designed to keep two one dimensional meshes a minimum distance apart from one another. The distance between centroid axes, which are represented by the finite element meshes, always remains positive and an overlap between beams corresponds to the distance dropping below a threshold value, e.g. $R^{+} + R^{-}$.

Three simplifications are currently used when performing the geometric calculations:

1. The centroid interpolation is linear even if higher order elements are being used. (The elements of the mesh are always line segments.)

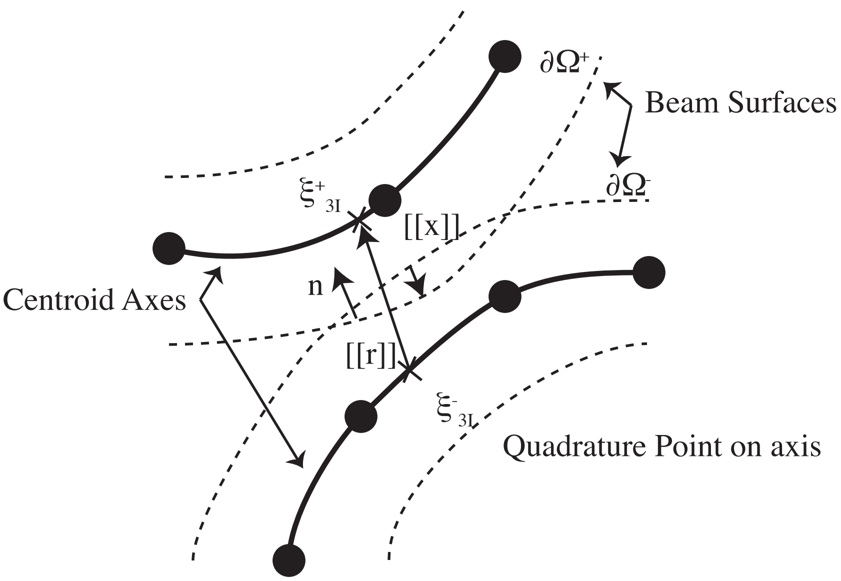


Figure 4.3.2: Contact geometry for beams

2. The cross sections do not deform and remain circular.
3. The contact area can be approximated with a cylinder-cylinder Hertzian response.

These simplifications were only made to reduce the computational cost and reduce the implementation complexity and are not a limitation of the beam formulation; it is possible to construct more detailed calculations for deformed higher-order beam surfaces. The first two simplifications enable an analytical solution to be found for one unknown, instead of a newton iteration being required to simultaneously solve for 5 unknowns (the three coordinates on the $-$ beam and the cross section coordinates on the surface of the $+$ beam.)

The closest-point mapping reduces to the equation

$$\xi^{I-} = \arg\min_{\xi_r^-} \left| \mathbf{r}^+ \left(\xi_3^{I+} \right) - \mathbf{r}^- \left(\xi_3^- \right) \right|. \tag{4.3.1}$$

When constructing the contact mapping, equally spaced points are placed along one element on $+$ beam, and the closest point on the $-$ beam is found.

The normal component at the surface of the two beams is calculated using only the centroid positions,

$$\mathbf{n} = \frac{[[\mathbf{r}]]}{||[[\mathbf{r}]]||_2}.$$

Using that assumption, the overlap in the kinematic displacement fields is defined just by the centroid positions:

$$\begin{aligned} [[\mathbf{x}]] &= \left(R^+ + R^- - \left(\mathbf{r}^+ \left(\xi_I^+ \right) - \mathbf{r}^- \left(\xi_I^- \right) \right) \cdot \mathbf{n} \right) \mathbf{n} \\ &= \left(2 \langle R \rangle - ||[[\mathbf{r}]]||_2 \right) \mathbf{n}. \end{aligned}$$

The beam meshes are only one dimensional, so the the integrals that will be evaluated will be one dimensional along the centroid of the beam. The integral of the jump condition over the entire contact area is required, so a differential line thickness a at a contact point ξ_3^+, ξ_3^- is required to transform the area integral into a one-dimensional integral:

$$\int_{\Gamma_C} \mathbf{f} \cdot \mathbf{n} dA = \int_{\xi_3 \in \Gamma_C} \mathbf{f} \cdot \mathbf{n} a d\xi_3$$

The mechanical analysis will be acceptable without considering the contact area because the penalty parameter is chosen to be arbitrary. Thus, the penalty condition can stated as

$$\int_{\Gamma_C} \mathbf{f} \cdot \mathbf{n} dA = \int_{\xi_3 \in \Gamma_C} P^* \left[\left[\mathbf{x}\right]\right] \cdot \mathbf{n} d\xi_3.$$

However, the thermal and electrical contacts do use jump conditions with physical origins, so the contact area is required for consistency. The dependence on contact area gives an implicit coupling between the mechanical fields and the thermal and electric contacts: increasing the contact pressure increases the contact area, which in turn decreases the contact resistances, even when not considering tribological effects. (Indeed, at the textile macroscale, this is a tribological effect.) Due to the arbitrary nature of the penalty, the overlap is nonphysical. The contact pressure, which is a physical quantity, can be used to calculate an appropriate contact area. Hertzian contact theory can be used as an adequate approximation using the case of two parallel cylinders [65]:

$$a = \sqrt{\frac{4PR^*}{\pi E^*}}.$$

where R^* and E^* are the effective radius and effective Young's modulus, respectively, between the two cylinders. These quantities are calculated by $E^* = \left(\frac{E^+}{1-\nu_+^2} + \frac{E^-}{1-\nu_-^2}\right)^{-1}$ and $R^* = \left(R_+^{-1} + R_-^{-1}\right)^{-1}$. The total differential load is denoted by P, i.e. the integral of the traction along a differential line element of the contact surface, $P\left(\xi_3\right) = \int_{d\Gamma_C(\xi_3)} \mathbf{t}\left(\xi_3\right) \cdot \mathbf{n} ds$. This is recognized as the penalty jump condition, $P\left(\xi_3\right) = P^* \left[\left[\mathbf{x}\right]\right] \cdot \mathbf{n}$, This enables the differential thickness to be written as

$$a = \sqrt{\frac{4R^*}{\pi E^*}\left(P^* \left[\left[\mathbf{x}\right]\right] \cdot \mathbf{n}\right)}.$$

Then, the differential contributions of the thermal and electrical jump conditions can be expressed along the differential line element $d\xi_3$,

$$\begin{aligned} h_C \left[\left[T\right]\right] dA &= h_C \left[\left[T\right]\right] \sqrt{\frac{4R^*}{\pi E^*}\left(P^* \left[\left[\mathbf{x}\right]\right] \cdot \mathbf{n}\right)} d\xi_3 \\ \sigma_C \left[\left[V\right]\right] dA &= \sigma_C \left[\left[V\right]\right] \sqrt{\frac{4R^*}{\pi E^*}\left(P^* \left[\left[\mathbf{x}\right]\right] \cdot \mathbf{n}\right)} d\xi_3, \end{aligned}$$

coupling them nonlinearly to the mechanical field.

4.3.4 Contact Mapping Generation

For each pair of contacting meshes, a discrete contact mapping needs to be constructed. The continuous contact zones are discretized as tables of quadrature points for nearby elements, illustrated in Figure 4.3.3. The discrete mapping is represented as a list of active

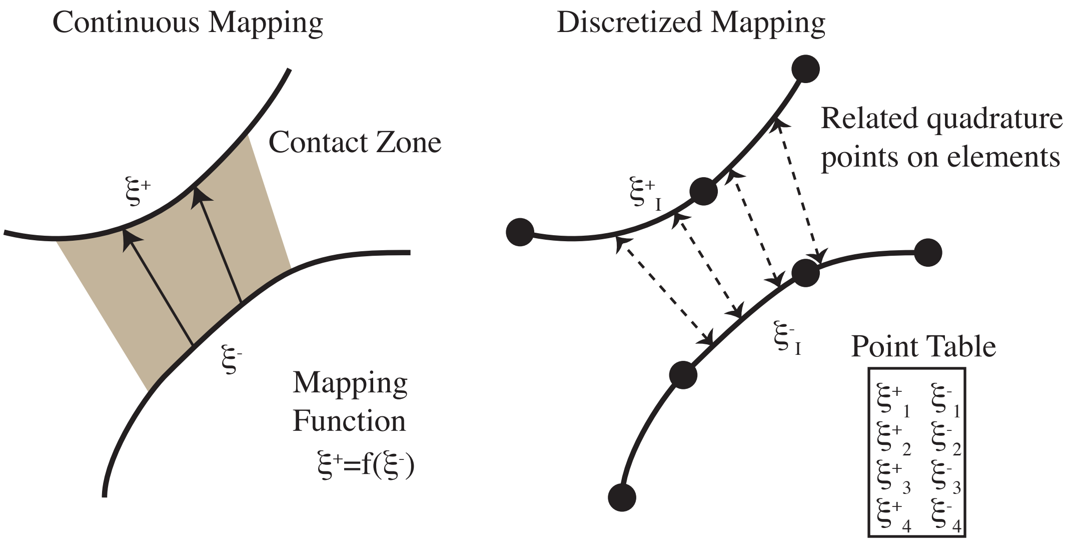


Figure 4.3.3: Discretized contact mapping

interacting element pairs and a corresponding quadrature point table. The quadrature point table for each active pair has $2N_{GP}$ entries: one list of points on the + element, and a corresponding list of points on the − element. The data structure of N_{active} element pairs with indices e_I^+ and e_I^- matching to a quadrature point table can be illustrated as:

$$\begin{array}{ccc} e_1^+ & e_1^- & \left\{\left(s_1^+, s_1^-\right), \left(s_2^+, s_2^-\right), \ldots \left(s_{N_{GP}}^+, s_{N_{GP}}^-\right)\right\}_1 \\ e_2^+ & e_2^- & \left\{\left(s_1^+, s_1^-\right), \left(s_2^+, s_2^-\right), \ldots \left(s_{N_{GP}}^+, s_{N_{GP}}^-\right)\right\}_2 \\ \vdots & & \\ e_{N_{active}}^+ & e_{N_{active}}^- & \left\{\left(s_1^+, s_1^-\right), \left(s_2^+, s_2^-\right), \ldots \left(s_{N_{GP}}^+, s_{N_{GP}}^-\right)\right\}_{N_{active}} \end{array}$$

The quadrature point table is determined by picking equally-spaced points along the + element and finding the closest point on the − element. The points need not be equally spaced and are only chosen to be so for simplicity; it is possible that there exists some scheme to more accurately and efficiently integrate this type of contact integral with unequally spaced points. Using linear shape functions, the displacement field along the element on the − beam can be described

$$\mathbf{r}^-(s) = \phi^1(s)\,\mathbf{r}_-^1 + \phi^2(s)\,\mathbf{r}_-^2 = s\mathbf{r}_-^1 + (1-s)\,\mathbf{r}_-^2$$

with $s \in [0,1]$. From Equation 4.3.1, the arg min problem with this discretization is

$$\begin{aligned} s_I^- &= \arg\min_{s^-}\left|\mathbf{r}_I^+ - s\mathbf{r}_-^1 - (1-s)\,\mathbf{r}_-^2\right| \\ &= \arg\min_{s^-}\left(\mathbf{r}_I^+ - \mathbf{r}_-^2 + s\left(\mathbf{r}_-^2 - \mathbf{r}_-^1\right)\right)\cdot\left(\mathbf{r}_I^+ - \mathbf{r}_-^2 + s\left(\mathbf{r}_-^2 - \mathbf{r}_-^1\right)\right) \\ &= \arg\min_{s^-}\left(\left(\mathbf{r}_I^+ - \mathbf{r}_-^2\right)\cdot\left(\mathbf{r}_I^+ - \mathbf{r}_-^2\right) + 2s\left(\mathbf{r}_-^2 - \mathbf{r}_-^1\right)\cdot\left(\mathbf{r}_I^+ - \mathbf{r}_-^2\right)\right. \\ &\qquad \left. + s^2\left(\mathbf{r}_-^2 - \mathbf{r}_-^1\right)\cdot\left(\mathbf{r}_-^2 - \mathbf{r}_-^1\right)\right). \end{aligned}$$

This equation can be solved analytically by taking the derivative of the of the minimization problem,

$$\left(\mathbf{r}_{-}^{2}-\mathbf{r}_{-}^{1}\right)\cdot\left(\mathbf{r}_{I}^{+}-\mathbf{r}_{-}^{2}\right)=s\left(\mathbf{r}_{-}^{2}-\mathbf{r}_{-}^{1}\right)\cdot\left(\mathbf{r}_{-}^{2}-\mathbf{r}_{-}^{1}\right)$$

to solve for the corresponding point

$$s_{I}^{-}=\frac{\left(\mathbf{r}_{-}^{2}-\mathbf{r}_{-}^{1}\right)\cdot\left(\mathbf{r}_{I}^{+}-\mathbf{r}_{-}^{2}\right)}{\left(\mathbf{r}_{-}^{2}-\mathbf{r}_{-}^{1}\right)\cdot\left(\mathbf{r}_{-}^{2}-\mathbf{r}_{-}^{1}\right)}. \tag{4.3.2}$$

A fibril assembly consists of many independent meshes which can all be contacting independently. One such list of element pairs exists for every interacting pair of meshes as well. The implementation and procedure details are explained in Section 5.5.

4.4 Variational Form

4.4.1 Function spaces

The continuum problem has four unknown fields: $\mathbf{x}, \mathbf{v}, T, V$. In the kinematic beam model, the fields $\mathbf{x}$ and $\mathbf{v}$ have assumed forms are not directly the functions being sought. The solutions that are sought are thus constructed from component functions of $\mathbf{r}$, $\mathbf{q}_1$ and $\mathbf{q}_2$. There are eight unknown fields

$$\mathbf{r},\ \mathbf{q}_1,\ \mathbf{q}_2,\ \mathbf{vr},\ \mathbf{vq}_1,\ \mathbf{vq}_2,\ T,\ V$$

with a total of twenty components. The test displacement and velocity fields constructed from test functions from the space of solutions of $\mathbf{r}$, $\mathbf{q}_1$, and $\mathbf{q}_2$, relating them to the test functions in the space of w by the relations

$$\begin{aligned}\delta\mathbf{x} &= \delta\mathbf{r}+\xi_1\delta\mathbf{q}_1+\xi_2\delta\mathbf{q}_2,\\ \delta\mathbf{v} &= \delta\mathbf{vr}+\xi_1\delta\mathbf{vq}_1+\xi_2\delta\mathbf{vq}_2.\end{aligned}$$

For the sake of clarity at the expense of compactness, the following naming convention for fields is adopted: prefix of δ denotes a test function, prefix of $\mathbf{v}$ denotes an element of the velocity field, and an overdot $\dot{a}$ denotes the time derivative, with an overline $\dot{\overline{abc}}$ used to prevent ambiguity with multi-character symbols.

When solving the problems individually, three separate functions are considered, w, T and V. The field w contains the three velocity fields. An additional antiderivative for the second-order mechanical fields is required, so let $x=\int w dt$. The three fields can be grouped together with their test functions:

$$\begin{aligned}x &= \left\{\begin{array}{c}\mathbf{r}\\ \mathbf{g}_1\\ \mathbf{g}_2\end{array}\right\},\ w=\left\{\begin{array}{c}\mathbf{vr}\\ \mathbf{vg}_1\\ \mathbf{vg}_2\end{array}\right\},\ \delta w=\left\{\begin{array}{c}\delta\mathbf{vr}\\ \delta\mathbf{vg}_1\\ \delta\mathbf{vg}_2\end{array}\right\}\\ & T,\delta T\\ & V,\delta V\end{aligned}$$

The position and velocity fields share the same test function since they are defined by the same function space.

For the monolithic scheme, single mixed function space w is constructed that contains all of the fields as

$$w = \left\{ \begin{array}{c} \left(\begin{array}{c} \mathbf{vr} \\ \mathbf{vg}_1 \\ \mathbf{vg}_2 \end{array} \right) \\ T \\ V \end{array} \right\}, \ \delta w = \left\{ \begin{array}{c} \left(\begin{array}{c} \delta\mathbf{vr} \\ \delta\mathbf{vg}_1 \\ \delta\mathbf{vg}_2 \end{array} \right) \\ \delta T \\ \delta V \end{array} \right\}$$

with corresponding test functions on each of those fields. The mechanical fields are grouped together with the parentheses. The position fields were left out because those will be handled specially. Consider now the derivative and antiderivative of the field w. Let $x = \int w dt$ and $\dot{w} = \frac{d}{dt} w$. These are represented in the function space by

$$x = \left\{ \begin{array}{c} \left(\begin{array}{c} \mathbf{r} \\ \mathbf{g}_1 \\ \mathbf{g}_2 \end{array} \right) \\ \emptyset \\ \emptyset \end{array} \right\}, \ \dot{w} = \left\{ \begin{array}{c} \left(\begin{array}{c} \dot{\overline{\mathbf{vr}}} \\ \dot{\overline{\mathbf{vg}_1}} \\ \dot{\overline{\mathbf{vg}_2}} \end{array} \right) \\ \dot{T} \\ \emptyset \end{array} \right\}.$$

The velocity fields are considered the primary fields so that the mechanical and thermal mass matrices are in the same form, $M\left(\delta w, \dot{w}\right)$. The nulls, $\emptyset$, are place holders in the finite element function space that are not included in variational form. These are here for convenience in the implementation and will leave behind zero rows in matrices. There is only a storage cost in the x vector associated with the null fields in the final program.

4.4.2 Equation

Without considering the fields from the beam formulations, the variational form of all of the equations considered is

$$\begin{aligned} 0 = \quad & \int_{\Omega_0} \delta\mathbf{v} \cdot \rho\dot{\mathbf{v}} + \delta T \left(\rho c_p + \frac{\partial \psi}{\partial T} \right) \dot{T} d\Omega \\ + \quad & \int_{\Omega_0} \nabla_X \delta\mathbf{v} : \mathbf{P}\left(\mathbf{x}\right) + \delta\mathbf{v} \cdot \mathbf{b} d\Omega + \oint_{\partial\Omega_0} \delta\mathbf{v} \cdot \mathbf{t} d\Gamma \\ + \quad & \int_{\Omega_0} \nabla_X \delta T \cdot \mathbf{k}_0 \nabla_X T + \delta T r d\Omega + \oint_{\partial\Omega_0} \delta T q_n d\Gamma \\ + \quad & \int_{\Omega_0} \nabla_X \delta V \cdot \epsilon_{r0} \nabla_X V - \nabla_X \delta V \cdot \epsilon_{r0} \mathbf{F}\mathbf{v} \times \mathbf{B} d\Omega + \oint_{\partial\Omega_0} \epsilon_{0r} \sigma_0^{-1} \mathbf{F}^T \boldsymbol{\mathcal{J}} \cdot \mathbf{N} d\Gamma \end{aligned}$$

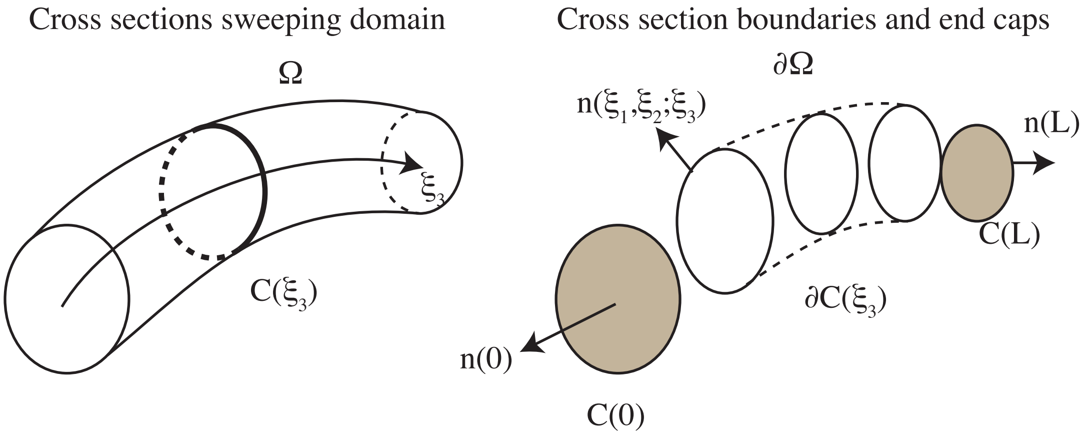

Cross sections sweeping domain
Ω
ξ3
C(ξ3)
Cross section boundaries and end caps
∂Ω
n(ξ1,ξ2;ξ3)
n(L)
C(L)
∂C(ξ3)
n(0)
C(0)

Figure 4.4.1: Decomposition of the beam domain

4.4.3 Incorporation of Contacts into Variational Form

Forces and fluxes are equal and opposite, so each of the variational forms on the two meshes have equal and opposite terms,

$$\begin{aligned}\mathbb{F}^{+} &= \int_{\Omega^{+}} a\left(\mathbf{u}^{+}\right) d\Omega ... + \int_{\Gamma_C} \delta\mathbf{u}^{+} \cdot \mathbf{f} d\Gamma \\ \mathbb{F}^{-} &= \int_{\Omega^{-}} a\left(\mathbf{u}^{-}\right) d\Omega ... + \int_{\Gamma_C} \delta\mathbf{u}^{-} \cdot -\mathbf{f} d\Gamma.\end{aligned}$$

The two forms can be added together, revealing the jump in the test function across the contact,

$$\begin{aligned}\mathbb{F} &= \int_{\Omega^{+}\cup\Omega^{-}} a\left(\mathbf{u}\right) d\Omega ... + \int_{\Gamma_C} \left(\delta\mathbf{u}^{+} - \delta\mathbf{u}^{-}\right) \cdot \mathbf{f} d\Gamma \\ &= \int_{\Omega^{+}\cup\Omega^{-}} a\left(\mathbf{u}\right) d\Omega ... + \int_{\Gamma_C} [[\delta\mathbf{u}]] \cdot \mathbf{f} d\Gamma.\end{aligned}$$

For heat conduction, the contact resistance has the integral:

$$\int_{\Gamma_C} [[\delta T]]\, q_n d\Gamma = \int_{\Gamma_C} [[\delta T]]\, h\, [[T]]\, d\Gamma$$

4.4.4 Integral Decomposition

Consider the variational form over a single fibril

$$0 = \int_{\Omega_\xi} a\left(\xi_1, \xi_2, \xi_3\right) d\Omega_\xi + \oint_{\partial\Omega_\xi} \mathbf{B}\left(\xi_1, \xi_2, \xi_3\right) \cdot \mathbf{n} d\Gamma_\xi.$$

The finite element discretization will only be applied to the centroid axis, ξ_3. Since assumed forms are used for the fields along the cross sections, the integrals will be split between the ξ_1, ξ_2 plane and the ξ_3 axis, as in Figure 4.4.1. Let $C(\xi_3)$ denote the cross section of the body in the reference configuration at position ξ_3 along the axis—a two dimensional open surface—and $\partial C(\xi_3)$ denote its boundary—a one dimensional closed loop. The volume can be obtained by sweeping the cross section along the centroid,

$$\Omega = \bigcup_{\xi_3 \in (0,L)} C(\xi_3).$$

The volume integrals can be separated into integrals along the length and integrals along the cross section:

$$\int_{\Omega_\xi} a(\xi_1, \xi_2, \xi_3)\, d\Omega_\xi = \int_0^L \int_{C(\xi_3)} a(\xi_1, \xi_2, \xi_3)\, dA_\xi d\xi_3.$$

The boundary of the domain can be separated into end caps at $\xi_3 = 0$ and $\xi_3 = L$ and the boundaries of the cross sections along the axis, such that

$$\partial\Omega_\xi = C(0) \cup C(L) \cup \bigcup_{\xi_3 \in (0,L)} \partial C(\xi_3).$$

The boundary integrals can then be broken up as

$$\begin{aligned} \oint_{\partial\Omega_\xi} \mathbf{B}(\xi_1, \xi_2, \xi_3) \cdot \mathbf{n} d\Gamma_\xi &= \int_{C(0)} \mathbf{B}(\xi_1, \xi_2, 0) \cdot \mathbf{n}(0)\, dA_\xi + \int_{C(L)} \mathbf{B}(\xi_1, \xi_2, \xi_3) \cdot \mathbf{n}(L)\, dA_\xi \\ &\quad + \int_0^L \oint_{\partial C(\xi_3)} \mathbf{B}(s, \xi_3) \cdot \mathbf{n}(s, \xi_3)\, ds d\xi_3 \end{aligned}$$

where s is the parametrization of the one dimensional contour such that the surface area element along the lengthwise surface of the beam is $dA = ds d\xi_3$. The unit normal of the end caps is equal to $\mathbf{N}_\xi = \pm\mathbf{E}_3$. Using these relations, the variational form can be written with a one-dimensional primary integral along the centroid direction,

$$\begin{aligned} &\underbrace{\int_0^L \left[\int_{C(\xi_3)} a(\xi_1, \xi_2, \xi_3)\, dA_\xi + \oint_{\partial C(\xi_3)} \mathbf{B}(s, \xi_3) \cdot \mathbf{n}(s, \xi_3)\, ds\right] d\xi_3}_{\text{Domain Integral}} \\ &\qquad + \underbrace{\left[\int_{C(\xi_3)} \mathbf{B}(\xi_1, \xi_2, \xi_3) \cdot \mathbf{n}(\xi_3)\, dA_\xi\right]_{\xi_3=0}^{\xi_3=L}}_{\text{Boundary Integral}} \end{aligned}$$

where the forms depend on integrals across the other directions.

4.4.5 Integral Discretization

The finite element approximation discretizes the solution along ξ_3, and the integral along $\int_0^L d\xi_3$ is handled by the finite element program. The ansatz aspect of the solution is integrated separately. Often times these integrals are handled analytically using common

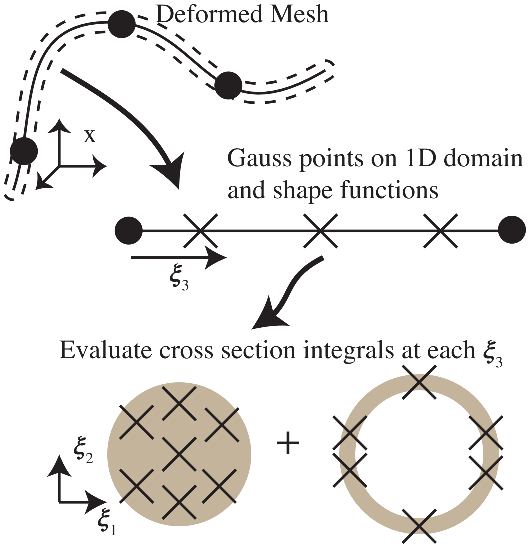

Deformed Mesh
x
Gauss points on 1D domain
and shape functions
ξ_3
Evaluate cross section integrals at each ξ_3
ξ_2
ξ_1
+

Figure 4.4.2: Splitting of integration between one dimensional finite elements along ξ_3 assumed cross section along ξ_1 and ξ_2

constitutive responses [103]. In this work, potentially complicated forms will be considered, so a more general scheme will be used. The cross section integrals are evaluated by Gaussian quadrature, so that each Gauss point on the finite element discretization maps to multiple Gauss points on the cross section and its boundary, illustrated in Figure 4.4.2. Let $w^A, \hat{\xi}^A \in P(\Omega)$ denote the set of Gauss points and weights over the geometric quantity Ω. The cross section integrals can be computed by the discrete summation

$$\int_{C(\xi_3)} a(\xi_1,\xi_2,\xi_3)\, dA_\xi = \sum_{P(C)} w^A a\left(\xi_3;\hat{\xi}_1^A;\hat{\xi}_2^A\right)$$

and the boundary integrals can be computed by

$$\begin{aligned} \oint_{\partial C(\xi_3)} \mathbf{B}(s,\xi_3)\cdot\mathbf{n}(s,\xi_3)\, ds &= \sum_{P(\partial C)} w^A b\left(\xi_3;\hat{\xi}_1^A;\hat{\xi}_2^A\right) \\ \int_{C(0)} \mathbf{B}(\xi_1,\xi_2,0)\cdot\mathbf{n}(0)\, dA_\xi &= \sum_{P(C)} w^A c\left(0;\hat{\xi}_1^A;\hat{\xi}_2^A\right). \end{aligned}$$

These forms can then be one-dimensional variational form can then be constructed

$$0 = \int_0^L \left[\sum_{P(C)} w^A a\left(\xi_3;\hat{\xi}_1^A;\hat{\xi}_2^A\right) + \sum_{P(\partial C)} w^A b\left(\xi_3;\hat{\xi}_1^A;\hat{\xi}_2^A\right)\right] d\xi_3 + \left[\sum_{P(C)} w^A c\left(\xi_3;\hat{\xi}_1^A;\hat{\xi}_2^A\right)\right]_0^L$$

that can be inputed into the finite element package as a one-dimensional integral.

4.4.6 Summed Variational Forms and Linearization

The variational form used to assemble the mass matrix for all the fibrils is

$$
\begin{aligned}
\mathbb{M} \;=\; & \sum_{f\in fib}\int_0^L \sum_I w^I \left(\delta\mathbf{vq}+\xi_1^I\delta\mathbf{vh}_1+\xi_2^I\delta\mathbf{vh}_2\right)\cdot\rho\left(\dot{\overline{\mathbf{vq}}}+\xi_1^I\dot{\overline{\mathbf{vh}_1}}+\xi_2^I\dot{\overline{\mathbf{vh}_2}}\right)d\xi_3 \\
& +\sum_{f\in fib}\int_0^L \sum_I w^I\delta T\dot{T}d\xi_3
\end{aligned}
$$

and the loading vector as a sum of all the fibril domains and contact surfaces is

$$
\begin{aligned}
\mathbb{F}\left(\{x_f,w_f\}\right) \;=\; & \sum_{f\in fib}\left[\int_0^L\left(\sum_{P(C)} w^A\left(\delta\mathbf{u}\cdot\frac{\delta\psi\left(\mathbf{u}\right)}{\delta\mathbf{u}}+\frac{d\delta T}{d\xi_3}\left(\frac{1}{\rho_0 c_p+\frac{\partial\psi}{\partial T}}\right)k_0\frac{dT}{d\xi_3}+\frac{d\delta V}{d\xi_3}\epsilon_{0r}\frac{dV}{d\xi_3}\right.\right.\right. \\
& \qquad +\delta\mathbf{u}\cdot\mathcal{J}\mathbf{e}_y\times\mathbf{B}+\delta T\frac{1}{\rho_0 c_p+\frac{\partial\psi}{\partial T}}\left(\sigma_0\mathcal{J}^2\right) \\
& \qquad \left.-\frac{d\delta V}{d\xi_3}\mathbf{E}_3\cdot\mathbf{F}^T\mathbf{v}\times\mathbf{B}\cdot\mathbf{E}_3\right) \\
& \qquad \left.+\sum_{P(\partial C)} w^A\left(\delta\mathbf{u}\cdot\bar{\mathbf{t}}+\delta T\bar{q}_n+\delta V\bar{D}_n\right)\right)d\Omega \\
& \qquad \left.+\left[\sum_{P(C)} w^A\left(\delta\mathbf{u}\cdot\bar{\mathbf{t}}+\delta T\bar{q}_n+\delta V\bar{D}_n\right)\right]_0^L\right] \\
& +\sum_{\Gamma\in C}\left(\int_{\xi_3\in\Gamma_C}\left[\left[\delta\mathbf{r}\right]\right]\cdot P^*\left(2\left\langle R\right\rangle-\left|\left[\left[\mathbf{r}\right]\right]\right|\right)\mathbf{n}\right. \\
& \qquad \left.+\left[\left[\delta T\right]\right]h\left[\left[T\right]\right]a+\left[\left[\delta V\right]\right]\sigma_C\left[\left[V\right]\right]ad\xi_3\right).
\end{aligned}
$$

With these two equations, the time dependent problem can be written compactly as

$$
\mathbb{M}\dot{w}+\mathbb{F}\left(x,w\right) \;=\; 0
$$

where $\mathbb{M}$ is the mass matrix and $\mathbb{F}$ is the force vector. To solve the problem with an implicit scheme, the linearizations of $\mathbb{F}$ with respect to x and w will be needed to product a system of the form

$$
\mathbb{M}\dot{w}+\mathbb{F}\left(x_0,w_0\right)+\left.\frac{\delta\mathbb{F}}{\delta x}\right|_{x_0,w_0}\Delta x+\left.\frac{\delta\mathbb{F}}{\delta w}\right|_{x_0,w_0}\Delta w=0
$$

is required. A time stepping scheme will be needed to discretize and relate w, $\dot{w}$, and x. The same linearizations can be used to solve static problems by setting w to 0,

$$
\mathbb{F}\left(x_0,0\right)+\left.\frac{\delta\mathbb{F}}{\delta x}\right|_{x_0}\Delta x=0.
$$

Chapter 5

Implementation Details

This section describes the details to the code used to implement the model and produce all of the results in this work. The incorporation of contacts in the finite element framework is described.

The code is dependent upon the finite element package FEniCS [85]. Its code generation features—especially automatic differentiation—make it an appealing choice. These features greatly facilitated the development of multiphysics beam formulations with implicit time stepping. The majority of the codebase used in this dissertation is a Python library for generating woven beam geometries and assembling their finite element forms. The library routines are designed to be used by short “main.py” script files to perform a specific calculation. The Python code uses a number of C++ extensions which are referenced as submodules.

The source code can be obtained from the git repositories hosted on the author’s bitbucket account, located at `https://bitbucket.org/afqueiruga/`. The supporting libraries and patches to FEniCS are available publicly. Access to the main library for the textile simulation can be granted by contacting the author. If the repositories are no longer accessible for whatever reason the future holds, contact the author to obtain a copy.

Additionally, a number of patches were made to the FEniCS code base to support contact mechanics, specifically Dolfin [87] and FFC [86]. These patches are necessary to run the program, and therefore the binary distributions will not be compatible. The patches can also be found alongside the other repositories at the bitbucket page. As it stands at the time of this writing, compilation of a custom build of FEniCS is required to run the program, which, in the author’s experiences, can be a daunting process. Upstreaming the features into the main FEniCS codebase is a future goal.

The code also makes use of Numpy [113] and Scipy [66] for the de facto standard Python numerical array data type and numerical algorithms; Mathematica [121] for code generation and symbolic integration and differentiation; Matplotlib [60] for plotting; and Paraview [56] for visualization. All graphs and images in this work were created using either Matplotlib or Paraview, with the exception of Figure 5.4.1, which was generated in Mathematica.

5.1 Finite Elements

A brief overview of the finite element method is described here in a non-pedagogical manner to provide context for the incorporation of contacts into the standard procedure and the place of code generation. The reader is directed to the following books on the finite element method: Hughes [58] for an introduction to the linear theory and Bonet [14] for a nonlinear treatment.

Functions are discretized to be linear combinations of a finite set of shape functions, with coefficients a_A weighting the corresponding shape function ϕ_A,

$$u\left(x\right) \approx \sum a_A \phi_A\left(x\right).$$

Test functions and trial functions in the variational calculus are also expressed as linear combinations of shape functions, such that $\delta u\left(x\right) = \sum b_A \phi_A\left(x\right)$ and $\Delta u\left(x\right) = \sum c_A \phi_A\left(x\right)$. The shape functions are picked from the same space as u, with the caveat that variations on Dirichlet boundaries must be zero, so the coefficients corresponding to those modes must be zero. The linear system to be solved is extracted from these coefficients. The arbitrariness of test functions translates into extracting rows of a matrix or vector from corresponding coefficients, i.e.

$$\int \delta u f d\Omega \, \forall \delta u \rightarrow b_A \int \phi_A f d\Omega \, \forall b_A \rightarrow R_A = \int \phi_A f d\Omega.$$

Trial functions represent a field that can be solved linearly, so those coefficients extract the columns of a matrix,

$$\int \delta u k \Delta u d\Omega \rightarrow b_A \int \phi_A k \phi_B d\Omega c_B \rightarrow K_{AB} c_B = \left[\int \phi_A k \phi_B d\Omega\right] c_B$$

that can be back-solved for c_B.

In nonlinear finite elements, the variational equation,

$$\int \delta u f\left(u\right) d\Omega = 0 \, \forall \delta u$$

becomes a system of nonlinear functions of coefficients $u = \sum a_B \phi_B$,

$$\int \phi_A f\left(\sum a_B \phi_B\right) d\Omega = 0 \, \forall A.$$

A trial function $\Delta u = \sum c_B \phi_B$ is picked of the same space of δu. The nonlinear equation is solved by linearizing the unknown function in an unknown direction,

$$\int \delta u f\left(u^I\right) d\Omega + \int \delta u \left.\frac{\partial f}{\partial u}\right|_{u^I} d\Omega \Delta u = 0 \, \forall \delta u$$

to make a corresponding linear system

$$\underbrace{\left\{\int \phi_A f\left(a^I_B \phi_B\right) d\Omega\right\}}_{R_A} + \underbrace{\left[\int \phi_A \left.\frac{\partial f}{\partial u}\right|_{a^I_B} \phi_B d\Omega\right]}_{K_{AC}} \left\{c_C\right\} = 0$$

that can be solved for the coefficients c_B. The coefficients to the unknown function u are then updated iteratively by $a_B^I + = c_B$ until the process converges.

In the finite element method, the domain is discretized into a mesh where the cells are referred to as elements, $\Omega = \bigcup_e \Omega^e$. Let e be used to denote individual elements, M to denote the mesh as a set of elements, and Ω^e to denote the domain on an individual element. Let N be the total amount of unknown degrees of freedom, and n be the degrees of freedom supported by a single element. This allows the integrals over domains to be broken up as

$$\int_\Omega f d\Omega = \sum_{e \in M} \int_{\Omega^e} f d\Omega.$$

The shape functions in the finite element method are constructed by piecewise additions of compactly supported functions on an element domain. The assembly of the linear system involves local contributions that only couple the coefficients associated with one element, i.e. only $n \times n$ or n sized matrices and vectors. The finite element procedure of placing smaller local element into a global matrix can be summarized

$$\underbrace{\mathbf{K}}_{N\times N} = \mathop{\mathsf{A}}_{e} \underbrace{[\mathbf{k}^e]}_{n\times n}$$

where A is an operator that represents the assembly program. The entries in the local matrix are placed in the global matrix based on the connectivity table entry of the element, $\texttt{dofs}\,[e]$. The table provides a mapping from local index a to global index A by $A = \texttt{dofs}\,[e]\,[a]$. Each step in the assembly involves the summation operation

$$K_{(\texttt{dofs}[e][a])(\texttt{dofs}[e][b])} + = k^e_{ab}\, \forall a, b \in [1, n]\,.$$

For the contact handling, the assembly operation happens multiple times on different meshes and different contacts. Let $\{M\}$ denote the set of meshes and contacts, and + and − denote the mesh indices for the two contacting meshes. The contact integrals represent couplings between the degrees of freedom associated with two different meshes, so the local contributions are of size $2n \times 2n$ or $2n$. There are two domains and two sets of shape functions in the local calculation, ϕ_A^+ and ϕ_A^-. The contact mapping is also discretized by the two meshes as lists of interacting element pairs and corresponding mappings. Denoting the individual pair-wise mappings as γ_c, the total contact surface is broken up as $\Gamma_C = \bigcup_c \gamma_c$, allowing the integral to also be calculated in components by

$$\int_{\Gamma_C} [[\phi_A]]\, [[\phi_B]]\, d\Gamma = \sum_c \int_{\gamma_c} [[\phi_A]]\, [[\phi_B]]\, d\Gamma.$$

The local matrix contribution for a single contact can be constructed from blocks,

$$\underbrace{[\mathbf{k}^c]}_{2n\times 2n} = \begin{bmatrix} \underbrace{\left[\int \phi_A^+ \phi_B^+ d\Gamma\right]}_{n\times n} & \underbrace{\left[\int \phi_A^+ \phi_B^- d\Gamma\right]}_{n\times n} \\ \underbrace{\left[\int \phi_A^- \phi_B^+ d\Gamma\right]}_{n\times n} & \underbrace{\left[\int \phi_A^- \phi_B^- d\Gamma\right]}_{n\times n} \end{bmatrix}$$

where each block represents the coupling from one element to either itself or the other. The total assembly process is now

$$\underbrace{\mathbf{K}}_{N\times N} = \sum_{M\in\{M\}} \mathop{\mathsf{A}}_{e\in M} \underbrace{\left[\mathbf{k}^{e}\right]}_{n\times n} + \sum_{\Gamma_C\in\{M\}} \mathop{\mathsf{A}}_{c\in\Gamma_C} \underbrace{\left[\mathbf{k}^{c}\right]}_{2n\times 2n} .$$

Each mesh has its own connectivity table, $\texttt{dofs}^{I}$ for mesh I, and the connectivity entries from both elements, $\texttt{dofs}^{+}\left[e^{+}\right]$ and $\texttt{dofs}^{-}\left[e^{-}\right]$, are needed to place the blocks into the global matrix. The assembly operation for each contact requires placing the four blocks,

$$\begin{aligned}
K_{(\texttt{dofs}^{+}[e^{+}][a])(\texttt{dofs}^{+}[e^{-}][b])} &\mathrel{+}= k_{ab}^{c++}\,\forall a,b\in[1,n]\\
K_{(\texttt{dofs}^{+}[e^{+}][a])(\texttt{dofs}^{-}[e^{-}][b])} &\mathrel{+}= k_{ab}^{c+-}\,\forall a,b\in[1,n]\\
K_{(\texttt{dofs}^{+}[e^{+}][a])(\texttt{dofs}^{-}[e^{-}][b])} &\mathrel{+}= k_{ab}^{c-+}\,\forall a,b\in[1,n]\\
K_{(\texttt{dofs}^{+}[e^{+}][a])(\texttt{dofs}^{-}[e^{-}][b])} &\mathrel{+}= k_{ab}^{c--}\,\forall a,b\in[1,n]\,.
\end{aligned}$$

The code for this procedure is described in Algorithm 5.1.

5.2 Modifications to FEniCS

There are two facets to a finite element code: (1) the calculation of the local finite element matrices and vectors and (2) the assembly of these local matrices and vectors into the global matrix or vector. The purpose of using a domain specific language is to assist with formulating and transcribing the equations for the calculation of the local matrices for the problem into computer code.

In its current state, circa 2014, FEniCS was only able to perform calculations on a single mesh. The contact formulation requires calculations across multiple different meshes. Prototype code for cut-and-composite meshes, now the MultiMesh set of objects that made it into version 1.5, existed in the development version and was extremely useful in developing the contact code. The patches made to dolfin and ffc, as well as extension modules for assembling, will be described in this section.

5.2.1 Form Language and Compiler

Using the UFL representation, FFC automatically generates a C++ object that contains, in addition to a lot of boiler-plate code for data management, shape function and quadrature point calculation, a `tabulate_tensor` routine that performs the local element matrix calculation. The form language has a few integral types that it can express: `dx` for domains, $\int d\Omega$; `ds` for exterior surfaces, $\oint_{\partial\Omega} d\Gamma$; and `dS` for interior element boundaries, $\sum_i \int_{S_i} dS$. There is also a custom integral, `dc`, which serves exactly the purpose of implementing nonstandard features. FEniCS contained syntax existed for expressing jump conditions for discontinuous fields in the form language, and supported local element matrices computed over two surfaces. These are used in the interior surface integrals (`dS`) for the discontinuous Galerkin method.

Taking advantage of the existing syntax, the only modification needed to the form language is an identifier to the custom integral identifier. This was accomplished by adding a “special” tag to the integral metadata. FFC was modified to look for this tag and trigger special compilation code based on the value of the tag. The only value with an assigned meaning is “contact”, but having this tag may facilitate the future implementation of experimental features. A snippet of the beam implementation illustrates the contact integral:

```
xr = X0 + q
dist = sqrt(dot(jump(xr),jump(xr)))
overlap = (2.0*avg(radius)-dist)
cont_pres = contact_penalty*overlap
ContactForm = -dot(jump(tvq),
    conditional(ge(overlap,0.0), -cont_pres,0.0)*jump(xr)/dist) \
    *dc(0, metadata={"num_cells": 2,"special":"contact"})
```

From the end user perspective, specifying the custom integral tag and the right metadata is the only thing required to generate the local element assembly code.

From the developer’s perspective (i.e., the author), the existing analysis and differentiation algorithms in FFC perform all of the expression manipulations required. Most of the new capabilities and modifications were centered in the code generation components. The assembler is required to feed custom integrals (those specified by `dc`) the list of quadrature points and weights, as opposed to the other integral types which will generate their own quadrature points inside of the tabulation routine. Even on two-cell calculations, the quadrature points would be shared between the two sets of shape functions because the original intentions were for elements sharing a facets or overlapping elements in cut-and-composite meshes. When the “special” : “contact” tag is encountered in code generation routine, it is now expected that the quadrature point list will be twice long, with the same number of quadrature weights, to accommodate the quadrature point-pairs described in the previous chapter. The basis functions and their derivatives for both cells receive different quadrature points by adding an offset to the index to the point table. In rough pseudocode, the evaluation for the two sets of shape functions at quadrature point I is

$$\begin{aligned}
\left\{\phi^{+}_{AI}\right\} &= \texttt{eval\_basis}\left(\text{points} + N_{Dim} I, \text{vertices}\right) \\
\left\{\phi^{-}_{AI}\right\} &= \texttt{eval\_basis}\left(\text{points} + N_{Dim} I + N_{Dim} N_{GP}, \text{vertices} + N_{Dim} N_{Vertex}\right)
\end{aligned}$$

where A is the node number index and the routine calculates all of the shape functions simultaneously. Pointer arithmetic was used in the above equation, so that the addition of lengths to an array denotes passing the evaluation routine the latter part of the arrays.

Most of the modifications take place in the file `ffc/quadrature/quadraturegenerator.py`, which handles the final generation step for custom integrals. Additional code snippets were added to handle the quadrature table offsets. Some modifications were made to `ffc/quadrature/quadraturetransformerbase.py` as well. Namely, the evaluation of finite element functions needed to use the correct coefficients when they were evaluated on two different cells. The coefficient table, `w`, is twice as long for a macro element, where the coefficient array for the $-$ element is simply concatenated to the end of those form the $+$

element. The formulae to calculate the discrete function on both elements are

$$\begin{aligned} f^{+}(x) &= \sum_{A} \mathtt{w}[A]\,\phi_A^{+}(x) \\ f^{-}(x) &= \sum_{A} \mathtt{w}[\mathtt{dim}+A]\,\phi_A^{-}(x) \end{aligned}$$

where `dim` is the number of coefficients, i.e. the number of shape functions, on the element in the discrete function space. These equations was added to the QuadratureTransformer class to trigger when the contact identification was seen in the integral metadata.

5.2.2 Assembly Code

The assembly code was written in external modules to minimize changes to the main code base during development. Only one new method was required to enable the assembly of two-mesh forms. The "UFC" object in dolfin is responsible for copying data from function spaces defined on the mesh over to local arrays to be ordered for the tabulation routine on the master element, e.g. copying coefficients and vertices. The "UFC.update()" routine has a call signature for copying the data from two elements into the local arrays, but because the "Form" objects are closely tied to a mesh and thus particular a degree-of-freedom connectivity, the "UFC" object is also closely tied to a mesh. An extra call signature for the "UFC.update()" routine was added that takes a second "UFC" object and uses it's data to copy into the first objects data arrays. When assembling over a contact pair, two "UFC" objects are initialized for the two "Form" objects on each mesh. The update routine is called on the first object, with the second "UFC" object as an argument. The first half of the data arrays on the first object are filled with the data from the first mesh, and the second half are filled by taking data from the second mesh. The first "UFC" object is then used for the local matrix tabulation.

5.3 File Hierarchy

The main library is organized into directories as follows:

- **src** Top level library source
  - **Forms** Modules containing UFL descriptions
  - **unit_tests** Script files for testing library functionality
  - **ContactMultiMesh** Contact mesh container class
  - **BroadcastAssembler** Contact group assembler class
  - **ProximityTree** Spatial tree class for finding nearby elements
  - **multiwriter** vtk file output class
- **programs** Collection of script files
  - **engineering_tests** Programs for verifying accuracy of the library
- **notes** Mathematica files used for code generation
- **post** Location for writing output files
- **data** Saved simulation state data

The user may need to make the post and data directories by executing "mkdir" before running a program as they are not included in the repository.

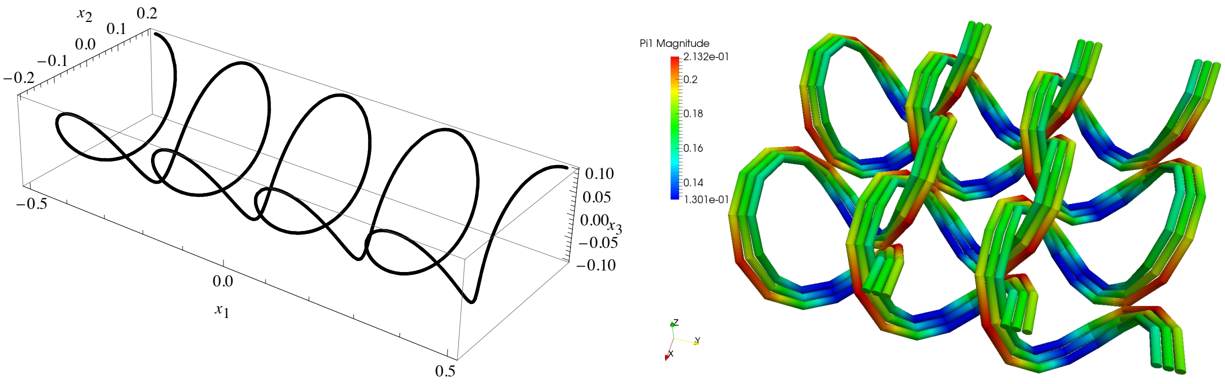


Figure 5.4.1: An example initial curve for a single yarn in a knit, designed with $\mathbf{y}\left(X_3\right) = \left(1+C\right)X_3 + B\sin pX_3\mathbf{e}_3 + A_1\cos\frac{p}{2}X_3\mathbf{e}_1 + A_2\cos pX_3\mathbf{e}_2$ where C is a uniformed "squishing", B is the forward-backward amplitude, A_1 and A_2 are the side-ways amplitudes, and p is the period of the looping. This curved is used for knitted textiles. An example RVE is shown on the right, made from three yarns each made from three fibrils.

5.4 Generation of initial configurations

The fabric and yarn structures studied have complicated intertwined geometries that that difficult to construct. For single strand structures, the mechanical process of spinning can be simulated during the relaxation procedure to arrive at the desired structure, as was done in [118], for example. This process could also be used for woven structures, but for complicated knits setting up the simulation would be too complicated. Instead, individual fibrils are directly initialized to some designed curve as an initial guess. The curves are designed to be simple sinusoidal function that exhibit the correct interlocking and knotting of the textile structure. The dynamic relaxation procedure is then applied to reach the static equilibrium, as the beams are placed in stressed configurations. An example of one of these curves for a knitted mesh is shown in Figure 5.4.1.

Where X_3 is the reference coordinate along the axis of the beam, the centroid displacement field is initially interpolated to a given parameterized curve $\mathbf{y}$,

$$\mathbf{q}\left(X_3, 0\right) = \mathbf{y}\left(X_3\right) - \mathbf{r}_0\left(X_3\right).$$

Without adjusting the directors, this process can create a self intersecting beam ($\det\mathbf{F} \leq 0$) if the curve $\mathbf{y}$ turns to sharply or back on itself. This is particularly an issue when generating knitted fabrics. The directors are set using the Frenet-Serret frame of the curve [51], using $\mathbf{y}''$ and $\mathbf{y}' \times \mathbf{y}''$, to create two mutually orthogonal directions to the curve, whose direction $\mathbf{y}'$. The designed curves are not necessarily of unit arclength, so the curvature vector (the second derivative of the curve) may not be perpendicular to the curve. This is corrected by creating a third orthogonal vector with $\mathbf{y}' \times \left(\mathbf{y}' \times \mathbf{y}''\right)$. The director displacements are thus set using the formulae

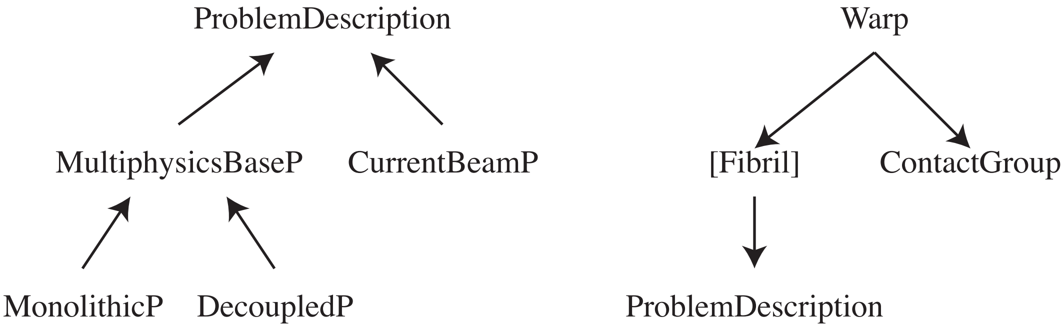


Figure 5.5.1: Inheritance diagram of Form objects (left) and data structure for mesh collections (right).

$$
\begin{aligned}
\mathbf{h}_1\left(X_3,0\right) &= R\frac{\mathbf{y}'\left(X_3\right)\times\mathbf{y}''\left(X_3\right)}{\left|\left|\mathbf{y}'\left(X_3\right)\times\mathbf{y}''\left(X_3\right)\right|\right|_2}-\mathbf{q}_{01}\left(X_3\right)\\
\mathbf{h}_2\left(X_3,0\right) &= R\frac{\mathbf{y}'\left(X_3\right)\times\left(\mathbf{y}'\left(X_3\right)\times\mathbf{y}''\left(X_3\right)\right)}{\left|\left|\mathbf{y}'\left(X_3\right)\times\left(\mathbf{y}'\left(X_3\right)\times\mathbf{y}''\left(X_3\right)\right)\right|\right|_2}-\mathbf{q}_{02}\left(X_3\right).
\end{aligned}
$$

The field produced by this process has torsion, depending on the curve, as the directors rotate to follow the curvature. However, it is not severe enough to cause any issues, and the beam will untwist as necessary during the dynamic relaxation procedure—yarns and fibrils are even expected to have a certain amount of twist. This method fails when $\mathbf{y}''$ is zero, but in this case the initial curve is straight and the directors can be set manually.

The curve $\mathbf{y}$ is designed in Mathematica and symbolic differentiation is used to create the forms for $\mathbf{h}_1$ and $\mathbf{h}_2$. The equations are then transformed into the C/Python code using Mathematica's CForm routine and some additional string manipulations to be copy-and-pasted into the script file.

5.5 Data structures and Algorithms

The variational forms described in the previous chapter are directly transcribed into UFL expressions with little hand-made optimization or simplification. The linearizations are directly made with the `derivative`() command. The "ProblemDescription" object contains dictionaries for sorting the material properties, function spaces, functions, and finite element form objects. This object is not specific to the textile simulation and was made for convenience. The object "CurrentBeamObject" extends the "ProblemDescription" object and contains the UFL code for only the mechanical beam formulation with a constant $\mathbf{I}\times\mathbf{B}$

body force. The object “MultiphysicsBaseProblem” implements the fully coupled thermal and electromagnetic problem. It also extends the “ProblemDescription” object, but is left a prototype by not implementing the function space definitions and form finalization. This is to maximize code reuse between the two different monolithic and decoupled function space formulations described in Section 4.4.1. The “MultiphysicsBaseProblem” object is extended twice by the “MonolithicProblem” and “DecoupledProblem” objects. Each object declares its function spaces, fields, and test and trial functions differently, which are then used by the parent class to create the same expressions. The expressions are returned to the child class to make different loading and linearization terms. The hierarchy is illustrated in Figure 5.5.1.

The primary object in the simulation is the Warp object, which represents a collection of Fibrils. Each Fibril object contains a corresponding problem description object, storing all of the function spaces, fields and Form objects defined on its mesh using dictionaries to store labels. The keys of the dictionaries on the individual meshes used to create corresponding MultiMesh objects in the Warp object. The Warp object also contains a ContactGroup object which calculates and stores the contact mapping data structures. Upon initialization, the Warp object creates a degree-of-freedom mapping to map all of the single-mesh function spaces into a larger function space to perform assemblies with multiple meshes. The Warp object performs the assembly of the each fibril and all of the contact pairs using the method described in Algorithm 5.1.

The contact data structure for the collection of meshes is a list is stored of the form $\left\{A, B, \left\{e_I^+, e_I^-, \left\{s_J^+, s_J^-\right\}\right\}\right\}$, where A and B are mesh indices with a corresponding list of active element pairs, which in turn has nested quadrature point tables. Automatically allocated Python lists are used for the topmost level of the nested lists since they can be of widely different sizes. The tables of quadrature points are stored more compactly using numpy arrays due to the larger amount of data stored. The list of meshes on which to perform the contact determination is specified at the beginning of the simulation based on the geometry of the textile weave pattern: e.g., possible pairs for an individual fibril are only fibrils in the same yarn and intersecting yarns. Due to the long, non-compact nature of the fibrils, this does not help significantly and reduces the number of mesh pairs by less than half for plain woven fabrics! When no such possible pair determination is possible or useful, an $\mathcal{O}\left(N_{mesh}^2\right)$ search is used. Though this type of search is less than ideal, N_{mesh} is small and on the order of 50 for most simulations. Spatial binary search trees are used to accelerate the identification of element pairs, so that the total search is on the order of $\mathcal{O}\left(N_{mesh}^2 N_{elem} \log N_{elem}\right)$. Future development will include implementing optimized contact search algorithms. The process is outlined in Algorithm 5.2.

5.6 Testing and Validation

Though the benchmarks are in Chapter 6 are the most interesting, additional tests are used to validate the program. The tests are split into two groups: unit tests and engineering tests. The purpose of the unit tests are to check program routines and algorithms for correctness. The submodules have unit tests located in their respective “___init___.py” files. The unit tests of the main library are located in “src/unit_tests/”.

Algorithm 5.1 Contact Form Assembly across multiple meshes

```
1. Let $g_N = \sum_{I \in meshes} N_I$ denote the total number of degrees of freedom.
2. Create an empty Matrix or Vector object of dimensions $g_N \times g_N$ or $g_N$.
3. For each mesh $M_I$: ($I$ is a mesh index)
   (a) Loop over all elements $e \in M_I$: ($e$ is element index)
         i. Insert the sparsity pattern for the form using the $I$th dof-map:
            $$\mathrm{dofs}_I\left(e\right) \otimes \mathrm{dofs}_I\left(e\right)$$
4. For each active pair $A, B$ in the contact group: ($A$ and $B$ are mesh indices)
   (a) Loop over active element pairs $\{e^+, e^-\}$:
         i. Insert sparsity pattern for the form using $A$th and $B$th dof-map:
            $$\left(\mathrm{dofs}_A\left(e\right) \cup \mathrm{dofs}_B\left(e\right)\right) \otimes \left(\mathrm{dofs}_A\left(e\right) \cup \mathrm{dofs}_B\left(e\right)\right)$$
5. Allocate memory for sparsity pattern
6. For each mesh $M^+$:
   (a) Loop over all elements $e \in M_I$:
         i. Collect field coefficients into local element array
        ii. Tabulate local element matrix
       iii. Add to global matrix
7. For each active pair $A, B$ in the contact group:
   (a) Loop over active element pairs $\{e^+, e^-\}$:
         i. Collect field coefficients from both meshes into macro element array
        ii. Collect gauss points into local table; compute weights as $1/N_{GP}$ (will be scaled
            by Jacobian in step iii)
       iii. Tabulate macro element matrix
        iv. Add to global matrix
8. Add all element contributions to finalize matrix
```

Algorithm 5.2 Contact Mapping Generation

```
1. Deform the reference meshes by the centroid displacement fields $\mathbf{q}$
2. Create binary search trees for the elements of each mesh
3. Create an empty list of active mesh pairs and respective contact mappings
4. For each pair of meshes $M^+, M^-$ with indices $A, B$ in the specified search table:
    (a) Create an empty contact mapping
    (b) For each element $e^+ \in M^+$:
        i. Use the binary search tree for $M^-$ to find all elements $e^-$ within the cutoff radius of $e^+$
        ii. For all of the found elements $e^-$:
            A. Create $N_{table}$ equally spaced points $s_I^+ \in [0, 1]$ and calculate $\mathbf{r}_I^+ = \mathbf{r}_1^+ \phi_1 \left( s_I^+ \right) + \mathbf{r}_2^+ \phi_2 \left( s_I^+ \right)$
            B. Use equation 4.3.2 for all $s_I^+$ to find corresponding $s_I^-$
            C. Compute the distances $\left| \mathbf{r}_I^+ - \mathbf{r}_I^- \right|$
            D. If any of the quadrature point pairs are within $1.2 \times (R^+ + R^-)$: add $e^+$, $e^- \left\{ s_I^+, s_I^- \right\}$ to the contact mapping; otherwise, discard element pair and gauss points.
    (c) If the contact mapping is not empty, add $A, B, \left\{ e^+, e^- \left\{ s_I^+, s_I^- \right\} \right\}$ to the list of active mesh pairs
```

The engineering tests verify the numerical accuracy of the program through convergence analysis and comparison to analytical solutions. The problems in Chapter 6 are two of these routines. Euler-Bernoulli beam problems are used using solutions from [45]. The source code for these tests can be found in "programs/engineering_tests.py". The electrical partial differential equation is verified by solving the potential drop for an undeformed fibril with a constant velocity and comparing against an analytical solution. The code is located in "programs/resistor_test.py".

5.7 Running a program

That standard procedure for a script file using the textile library is shown in Algorithm 5.3. The python source code for an example script file is included in Appendix A.

Algorithm 5.3 Procedure for a dynamic simulation

```
1. Create Geometry object and get end points
2. Define property dictionaries
3. Define element number array
4. Initialize Warp object on a ProblemDescription class
5. Apply Geometry initialize routine to Warp object, or load state data from file
6. Get Geometry objects mesh-pair list for contact search
7. Define boundary conditions and application routine
8. Define assembly function
9. Create DIRK object
10. Loop for $t \in 1, N_t$
    (a) Recalculate contact mappings
    (b) Perform RK timestep
    (c) Output data files
```

Chapter 6

Analysis of Numerical Solution Techniques

6.1 Introduction

The problem solved in Chapter 3 is used as a benchmark to verify and analyze the current-carrying beam formulation. The differential algebraic equation of the coupled second order, first order, and implicit PDEs is solved using a Diagonally Implicit Runge-Kutta method. The problem is solved both dynamically and statically using dynamic relaxation. The theoretical and numerical framework developed is verified by checking the order of convergence for different automatically generated finite element shape functions and compared to the analytical solution from Chapter 3.

6.2 Time stepping

With the implicit electromagnetic problem, it is not possible to use a purely explicit time marching scheme. This leaves three options for solving the state of the system at the next time step:

1. Explicit/Implicit splitting: march the dynamics explicitly, solving the implicit problems at each time step;
2. Implicit time stepping with iterative solution: solve each field implicitly, and iterate through the fields one by one until the solution converges;
3. Implicit time stepping with monolithic solution: solve all of the fields simultaneously with a single nonlinear newton iteration.

The first of these methods was used in previous work for a network model in [101]. The methodology is similar to solving the pressure-velocity differential algebraic equation for explicit solutions to the incompressible Navier-Stokes [48]. The second is employed in a similar network model in [130, 131]. The quasistatic analyses in Chapter 7 will be formulated using separated problems in this fashion. The third method will be used for the dynamic analysis in this chapter and Chapter 8.

6.2.1 Full linearization of a Diagonally Implicity Runge-Kutta (DIRK) method for a system with first order, second order and quasistatic components

Consider the following general system of differential equations that contains fully coupled second order ($\mathbf{x}(t)$ and $\mathbf{v}(t)$), first order ($T(t)$), and quasistatic ($V(t)$) components:

$$\begin{aligned}\frac{d}{dt}\mathbf{x} &= \mathbf{v}\\ \frac{d}{dt}\mathbf{v} &= f(\mathbf{x},\mathbf{v},T,V)\\ \frac{d}{dt}T &= g(\mathbf{x},\mathbf{v},T,V)\\ 0 &= e(\mathbf{x},\mathbf{v},T,V)\end{aligned}$$

Initial conditions of $\mathbf{x}(t=0)$, $\mathbf{v}(t=0)$, and $T(t=0)$ are required to fully define the problem. The initial quasistatic field $V(t=0)$ is not required because it must satisfy the equation $0 = e(\mathbf{x}(0), \mathbf{v}(0), T(0), V(0))$. The explicit time dependence is omitted in the following discussion for brevity, but let it be noted that f, g, and e may all be functions of time. Consider the case where the second order problem is a structural problem, where $\mathbf{x}$ and $\mathbf{v}$ are each vector fields, and the first order and quasistatic component are both scalar fields. The fields to have been discretized already so that $\mathbf{x}$, $\mathbf{v}$, T and V are treated arrays of discrete values. In this discussion, it does not matter what method was used to discretize the PDEs; the arrays are coefficients to finite element function spaces in this work. The total amount of degrees of freedom to be marched in time is $3N + 3N + 1N + 1N = 8N$. The majority of the degrees of freedom are associated with the discretization of $\mathbf{x}$ and $\mathbf{v}$, so is therefore desirable to treat the second order component specially to decrease the size of the matrix system to be solved (from $8N$ to $5N$ in the given example).

Diagonally implicit Runge-Kutta schemes are used, allowing the nonlinear problem of each stage to solved independently with only N-unknowns. The Butcher Tableaus of the methods are listed in Appendix C. The schemes used are all L-stable, meaning that certain modes in the system are damped to zero as their eigenvalue magnitude tends towards infinity[80]. The extra dampening of these methods is a desirable trait when performing dynamic relaxation for ill-behaved problems. Additionally, the final stage in these methods is at $t+\Delta t$ (i.e., $c_s = 1$) and the final row in A_{ij} is equal to b_j[1], and so the nonlinear solution step gives $x(t+\Delta t)$, $T(t+\Delta t)$, and $V(t+\Delta t)$ without any extra work. Normally when $c_s \neq 1$, the mass matrix can be back-solved with the linear combination to determine the values at $t+\Delta t$, but the quasi-static field would require performing an extra solution step. Because there is no time derivative—i.e. $V(t+\Delta t)$ is not actually required to calculate the next step—the quasi-static field could simply be left “dirty” and only solved when probing it is desired. For this work, only the L-stable methods will be used for the practical reasons described.

To illustrate the structure of the method, the Butcher Tableau of a general three stage DIRK is as follows:

[1] This property actually implies L-stability; see [1] for a proof. (The stability type was referred to as S-stability in that work; the nomenclature used by Leveque [80] is used here.)

$$\begin{array}{c|ccc} c_1 & a_{11} & 0 & 0 \\ c_2 & a_{21} & a_{22} & 0 \\ c_3 & a_{31} & a_{32} & a_{33} \\ \hline & b_1 & b_2 & b_3 \end{array}$$

For the rest of this section, h will be used to represent the time step size in place of Δt. For general Runge-Kutta methods marching a first-order ODE $\frac{d\mathbf{u}}{dt} = f\left(\mathbf{u}\right)$, the intermediate stage derivatives, $\mathbf{k}_i$ and the updated value $\mathbf{u}^{N+1}$ at $t+h$ at the are calculated by

$$\mathbf{k}_i = f\left(u_0 + h\sum_{j=1}^{s} a_{ij}\mathbf{k}_j\right)$$

and

$$\mathbf{u}^{N+1} = \mathbf{u}_0 + h\sum_{i=1}^{s} b_i\mathbf{k}_i.$$

The intermediate stage values are related to one another by

$$\mathbf{u}_i = \mathbf{u}_0 + h\sum_{j=1}^{i} a_{ij}\mathbf{k}_j = \mathbf{u}_0 + h\sum_{j=1}^{i} a_{ij} f\left(\mathbf{u}_j\right).$$

Now the multi-ordered system will be considered. For convenience, let the following symbols represent the current stage value minus the diagonal component (i.e., the unknown term): $\hat{\mathbf{v}}_i = \mathbf{v}_0 + h\sum_{j<i} a_{ij} f_j$, $\hat{\mathbf{x}}_i = \mathbf{x}_0 + h\sum_{j<i} a_{ij}\mathbf{v}_j$, and $\hat{T}_j = T_0 + h\sum_{j<i} a_{ij} T_j$. The second order system has a trivial equation $\frac{d\mathbf{x}}{dt} = \mathbf{v}$. Marching the trivial component first, each stage of the position is related to the previous stage values and current stage value of the velocity by

$$\mathbf{x}_i = \mathbf{x}_0 + h\sum_{j<i} a_{ij}\mathbf{v}_j + ha_{ii}\mathbf{v}_i = \hat{\mathbf{x}}_i + ha_{ii}\mathbf{v}_i.$$

This solution will be plugged in directly into the other equations to eliminate the need to solve for x_i independently. For the other equations in the system to be satisfied for the current stage, $\mathbf{v}_i$, T_i and V_i must satisfy

$$\begin{aligned} \mathbf{v}_i &= \mathbf{v}_0 + h\sum_{j=1}^{i} a_{ij} f\left(\mathbf{x}_0 + h\sum_{k=1}^{j} a_{jk}\mathbf{v}_k, \mathbf{v}_j, T_j, V_j\right) \\ T_i &= T_0 + h\sum_{j=1}^{i} a_{ij} g\left(\mathbf{x}_0 + h\sum_{k=1}^{i} a_{ik}\mathbf{v}_k, \mathbf{v}_j, T_j, V_j\right) \\ 0 &= e\left(\mathbf{x}_0 + h\sum_{k=1}^{i} a_{ik}\mathbf{v}_k, \mathbf{v}_j, T_j, V_j\right) \end{aligned}$$

Breaking the stage summations into the diagonal and lower triangular components to group together terms that depend on the current stage, the nonlinear system of equations

$$\begin{aligned} \mathbf{v}_i - ha_{ii} f\left(\hat{\mathbf{x}}_i + ha_{ii}\mathbf{v}_i, \mathbf{v}_i, T_i, V_i\right) &= \mathbf{v}_0 + h\sum_{j<i} a_{ij} f\left(\mathbf{x}_j, \mathbf{v}_j, T_j, V_j\right) \\ \mathbf{u}_i - ha_{ii} g\left(\hat{\mathbf{x}}_i + ha_{ii}\mathbf{v}_i, \mathbf{v}_i, T_i, V_i\right) &= \mathbf{u}_0 + h\sum_{j<i} a_{ij} g\left(\mathbf{x}_j, \mathbf{v}_j, T_j, V_j\right) \\ e\left(\hat{\mathbf{x}}_i + ha_{ii}\mathbf{v}_i, \mathbf{v}_i, T_i, V_i\right) &= 0 \end{aligned}$$

is obtained. Using the chain rule, the linearization of one the functions on the left hand side with respect to $\mathbf{v}_i$ is

$$\frac{\partial}{\partial \mathbf{v}_i} = ha_{ii}\frac{\partial}{\partial \mathbf{x}} + \frac{\partial}{\partial \mathbf{v}}.$$

The stage variables are linearized simultaneously around an iteration value indexed by m by $\left(\mathbf{v}_i^{[m]}, T_i^{[m]}, V_i^{[m]}\right) + \left(\Delta\mathbf{v}_i^{[m]}, \Delta T_i^{[m]}, \Delta V_i^{[m]}\right)$. The three systems are solved simultaneously using the incremental updates

$$\begin{aligned}
\left[\mathbf{I} - ha_{ii}\left(ha_{ii}\frac{\partial f}{\partial \mathbf{x}}^{[m]} + \frac{\partial f}{\partial \mathbf{v}}^{[m]}\right)\right]\Delta\mathbf{v} - ha_{ii}\frac{\partial f}{\partial T}^{[m]}\Delta T - ha_{ii}\frac{\partial f}{\partial V}^{[m]}\Delta V& \\
= \mathbf{v}_0 + h\sum_{j<i} a_{ij} f_j - \mathbf{v}_i^{[m]} + ha_{ii} f_i^{[m]}& \\
-ha_{ii}\left(ha_{ii}\frac{\partial g}{\partial \mathbf{x}}^{[m]} + \frac{\partial g}{\partial \mathbf{v}}^{[m]}\right)\Delta\mathbf{v} + \left(\mathbf{I} - ha_{ii}\frac{\partial g}{\partial T}^{[m]}\right)\Delta T - ha_{ii}\frac{\partial g}{\partial V}^{[m]}\Delta V& \\
= T_0 + h\sum_{j<i} a_{ij} g_j - T_i^{[m]} + ha_{ii} g_i^{[m]}& \\
\left(ha_{ii}\frac{\partial e}{\partial \mathbf{x}}^{[m]} + \frac{\partial e}{\partial \mathbf{v}}^{[m]}\right)\Delta\mathbf{v} + \frac{\partial e}{\partial T}^{[m]}\Delta T + \frac{\partial e}{\partial V}^{[m]}\Delta V = -e_i^{[m]},&
\end{aligned}$$

where $\mathbf{v}_i^{[m]}$, $f_i^{[m]}$, $\frac{\partial f}{\partial \mathbf{v}}^{[m]}$, etc. are the values and function evaluations at the current Newton step. Rewriting this as a block-matrix system, each step m of the Newton iteration requires solving the problem

$$\begin{bmatrix}
\left[\mathbf{I} - ha_{ii}\left(ha_{ii}\frac{\partial f}{\partial \mathbf{x}}^{[m]} + \frac{\partial f}{\partial \mathbf{v}}^{[m]}\right)\right] & \left[-ha_{ii}\frac{\partial f}{\partial T}^{[m]}\right] & \left[-ha_{ii}\frac{\partial f}{\partial V}^{[m]}\right] \\
\left[-ha_{ii}\left(ha_{ii}\frac{\partial g}{\partial \mathbf{x}}^{[m]} + \frac{\partial g}{\partial \mathbf{v}}^{[m]}\right)\right] & \left[\mathbf{I} - ha_{ii}\frac{\partial g}{\partial T}^{[m]}\right] & \left[-ha_{ii}\frac{\partial g}{\partial V}^{[m]}\right] \\
\left[ha_{ii}\frac{\partial e}{\partial \mathbf{x}}^{[m]} + \frac{\partial e}{\partial \mathbf{v}}^{[m]}\right] & \left[\frac{\partial e}{\partial T}^{[m]}\right] & \left[\frac{\partial e}{\partial V}^{[m]}\right]
\end{bmatrix}
\begin{Bmatrix} \Delta\mathbf{v} \\ \Delta T \\ \Delta V \end{Bmatrix}
= \begin{Bmatrix} \hat{\mathbf{v}}_i - \mathbf{v}_i^{[m]} + ha_{ii} f_i^{[m]} \\ \hat{T}_i - T_i^{[m]} + ha_{ii} g_i^{[m]} \\ -e_i^{[m]} \end{Bmatrix}$$

REMARK: Mass Matrices

Typically a mass matrix is present in the system of ordinary differential equations as a result of the spatial discretization, leading to equations of the form

$$\begin{aligned}
\dot{\mathbf{x}} &= \mathbf{v} \\
\mathbf{M}\dot{\mathbf{v}} &= f\left(\mathbf{x}, \mathbf{v}, T, V\right) \\
\mathbf{N}\dot{T} &= g\left(\mathbf{x}, \mathbf{v}, T, V\right) \\
0 &= e\left(\mathbf{x}, \mathbf{v}, T, V\right).
\end{aligned}$$

From a finite element representation, the matrices $\mathbf{M}$ and $\mathbf{N}$ result from an assembly of the form

$$M_{(Ai)(Bj)} = \int_\Omega \phi^A \rho \phi^B \delta_{ij} d\Omega$$

and

$$N_{AB} = \int_\Omega \phi^A \rho c_p \phi^B d\Omega,$$

in the case where $\mathbf{v}$ is a vector field and T is a scalar field. The subscript notation (Ai) refers to the index mapping calculation $3A+i$ for the degree of freedom index A and spatial coordinate index i. The system can be modified to incorporate the mass matrices as so:

$$\begin{bmatrix} \left[\mathbf{M} - ha_{ii}\left(ha_{ii}\frac{\partial f}{\partial \mathbf{x}}^{[m]} + \frac{\partial f}{\partial \mathbf{v}}^{[m]}\right)\right] & \left[-ha_{ii}\frac{\partial f}{\partial T}^{[m]}\right] & \left[-ha_{ii}\frac{\partial f}{\partial V}^{[m]}\right] \\ \left[-ha_{ii}\left(ha_{ii}\frac{\partial g}{\partial \mathbf{x}}^{[m]} + \frac{\partial g}{\partial \mathbf{v}}^{[m]}\right)\right] & \left[\mathbf{N} - ha_{ii}\frac{\partial g}{\partial T}^{[m]}\right] & \left[-ha_{ii}\frac{\partial g}{\partial V}^{[m]}\right] \\ \left[ha_{ii}\frac{\partial e}{\partial \mathbf{x}}^{[m]} + \frac{\partial e}{\partial \mathbf{v}}^{[m]}\right] & \left[\frac{\partial e}{\partial T}^{[m]}\right] & \left[\frac{\partial e}{\partial V}^{[m]}\right] \end{bmatrix} \begin{Bmatrix} \Delta\mathbf{v} \\ \Delta T \\ \Delta V \end{Bmatrix}$$

$$= \begin{Bmatrix} \hat{\mathbf{v}}_i - \mathbf{v}_i^{[m]} + ha_{ii} f_i^{[m]} \\ \hat{T}_i - T_i^{[m]} + ha_{ii} g_i^{[m]} \\ -e_i^{[m]} \end{Bmatrix}$$

6.2.2 Assembly via Monolithic Finite Element Forms

For the monolithic scheme, single mixed function space w is constructed that contains all of the fields:

$$w = \begin{Bmatrix} \begin{pmatrix} \mathbf{vr} \\ \mathbf{vg}_1 \\ \mathbf{vg}_2 \end{pmatrix} \\ T \\ V \end{Bmatrix}, \quad \delta w = \begin{Bmatrix} \begin{pmatrix} \delta\mathbf{vr} \\ \delta\mathbf{vg}_1 \\ \delta\mathbf{vg}_2 \end{pmatrix} \\ \delta T \\ \delta V \end{Bmatrix}$$

with corresponding test functions on each of those fields. The position fields were left out because those will be handled specially. Consider now the derivative and antiderivative of the field w. Let $x = \int w dt$ and $\dot{w} = \frac{d}{dt}w$. These are represented in the function space by

$$x = \begin{Bmatrix} \begin{pmatrix} \mathbf{r} \\ \mathbf{g}_1 \\ \mathbf{g}_2 \end{pmatrix} \\ \emptyset \\ \emptyset \end{Bmatrix}, \dot{w} = \begin{Bmatrix} \begin{pmatrix} \dot{\overline{\mathbf{vr}}} \\ \dot{\overline{\mathbf{vg}_1}} \\ \dot{\overline{\mathbf{vg}_2}} \end{pmatrix} \\ \dot{T} \\ \emptyset \end{Bmatrix}.$$

The nulls, $\emptyset$, are place holders in the finite element function space that are not included in variational form. These are here for convenience in the implementation and will leave behind zero rows in matrices. There is only a storage cost in the x vector associated with the null fields in the final program.

In this mixed-function space form, the system is compactly written as

$$M\dot{w} = f\left(x, w\right).$$

Following the same methodology as in the written-out case above, applying the scheme to x and plugging $x_i = x_0 + h\sum_{j<i} a_{ij}w_j + ha_{ii}w_i$ into equation for w_i yields a similar form of

$$Mw_i - ha_{ii}f\left(w_i, x_0 + h\sum_{k=1}^{i} a_{ik}w_k + ha_{ii}w_i\right) = Mw_0 + h\sum_{j=1}^{i-1} a_{ij}f\left(w_j, x_j\right).$$

The entire system can be linearized at once about $w_i^{[m]}$ in the direction Δw to make the Newton iteration:

$$\begin{aligned}\left(M - ha_{ii}\frac{\partial f}{\partial w} - h^2a_{ii}^2\frac{\partial f}{\partial x}\right)\Delta w &= Mw_0 + h\sum_{j=1}^{i-1} a_{ij}f\left(w_j, x_j\right) \\ &\quad -Mw_i^{[m]} + a_{ii}f\left(w_i^{[m]}, x_i^{[m]}\right).\end{aligned}$$

The rows of M and $\frac{\partial f}{\partial x}$ corresponding to Ø field placeholders are zero after the calculations are carried out. The linearized parts of the quasistatic equations appear in $\frac{\partial f}{\partial w}$. The system that is produced by this formulation of the linearizations is equivalent to the above system. The only different is the ordering of the unknowns: the finite element implementation is now free to automatically order the unknown fields. This results in a $\{x, y, z, T, V, x, y, z, T, V, ...\}$ ordering instead of a $\{x, x, ..., y, y, ..., z, z, ..., T, T, ..., V, V, ...\}$ ordering, greatly reducing the bandwidth of the linear systems.

6.3 Problem 1: Convergence of Static Analysis

In Chapter 3 it was observed that the wire has multiple static solutions corresponding to the circularity of the beam. The initial straight configuration is a bifurcation point, from which point multiple solution points are possible. Static analysis fails at this point. Dynamic relaxation is used as a starting point to guide the solution towards a valid solution, followed by a static analysis to finalize the solution. The velocity dependence prevents—or at least mediates—convergence issues caused by the onset of contacts and the lack of stiffness normal to the plane of the fabric and axis of the beams. Dynamic relaxation has been used in to study membranes [55]. Further, because of the very large deformations and steep contact penalties, the problem is not smooth enough to solve with standard nonlinear static analysis. An external dissipation force density is applied to the beam to bring it towards a steady state, equal to

$$\mathbf{f} = -\gamma\mathbf{v}$$

where γ is positive and large enough to make the motion critically damped.

The straight line will yield the solution in tension of the lowest circularity. To prevent bifurcation to other circularities due to buckling as the endpoints are brought inwards, a slight cosine shape is given to the z-direction displacement, as illustrated in the first frame in Figure 6.3.1. The initial velocity of the beam is chosen to bring the endpoints from their

Table 6.1: Static problem parameters

$\mathcal{J}$	$1A$	Δ	$\frac{L}{2}$
B	$1T$	r	0.02
ϕ	$\frac{\pi}{4}$	k	10
E	10	U	1.047

original points at $\pm L\mathbf{e}_1$ to $\pm\Delta\mathbf{e}_1$ after T_{max} time has passed to produce a gradual buckling process. This is satisfied by the initial condition

$$\mathbf{vq}\left(\mathbf{X}\right) = -\frac{L-\Delta}{T_{max}}X_1\mathbf{e}_1.$$

The boundary condition $\Delta\mathbf{vq} = \mathbf{0}$ is applied at both ends to specify that the Newton update is zero at each stage iteration on the velocities at the boundaries. It is possible to seek the other solutions by using initial conditions that differ from the straight line, such as helices of a higher desired circularity. Compressive solutions could also be sought with some more effort to set up initial conditions.

The time series of the solution process is shown in Figure 6.3.1. Most of the computational effort is spent during the dynamic relaxation phase, in fact, as about fifty time steps are needed before the static solution can begin. The penultimate frame is the final step of the dynamic relaxation phase, and the ultimate is the final step of the static solution. Using exactly the same finite element routines used to develop the finite element problem, the Newton iteration for the static analysis is

$$\mathbb{F}\left(x^I,0\right) + \frac{\delta\mathbb{F}\left(x^I,0\right)}{\delta x}\Delta x = 0$$

using the position at the final time $x\left(T\right)$ as the initial value x^0 where $x^{I+1} = x^I + \Delta x$ is the update for iteration step I. The velocities must be explicitly set to zero at the end of the dynamic relaxation phase due to the force law being dependent on the velocity (failing to do so changes the loading.)

To compare with the analytical solution, it is necessary to take the limit as the beam becomes more stringlike, or as the second moment of area approaches zero. For beam of circular section, this quantity is

$$I_{MOA} = \frac{\pi}{4}r^4.$$

This requires taking the limit as the radius decreases, $r \rightarrow 0$. To prevent the problem from scaling differently or even becoming unsolvable, the non-dimensional parameter $k = {}^{IB}/_{EA}$ is kept fixed. Picking a constant current density J (such that $I = JA$) instead of picking a total current satisfies this requirement. Thus, the analytical solution remains fixed as the radius is decreased. Additionally, the analytical solution was based on a linear elastic constitutive law. To keep the large-deformation constitutive law used in the numerical model in line with this assumption, k has to be kept small enough for the solution to be in the small-strain regime.

The parameters chosen are stated in Table 6.1. The $y = \mathbf{x}_2$ position of the center point of the beam, $l = 0$, $\mathbf{X} = 0, 0, 0$, is chosen to compare solutions. The process of decreasing

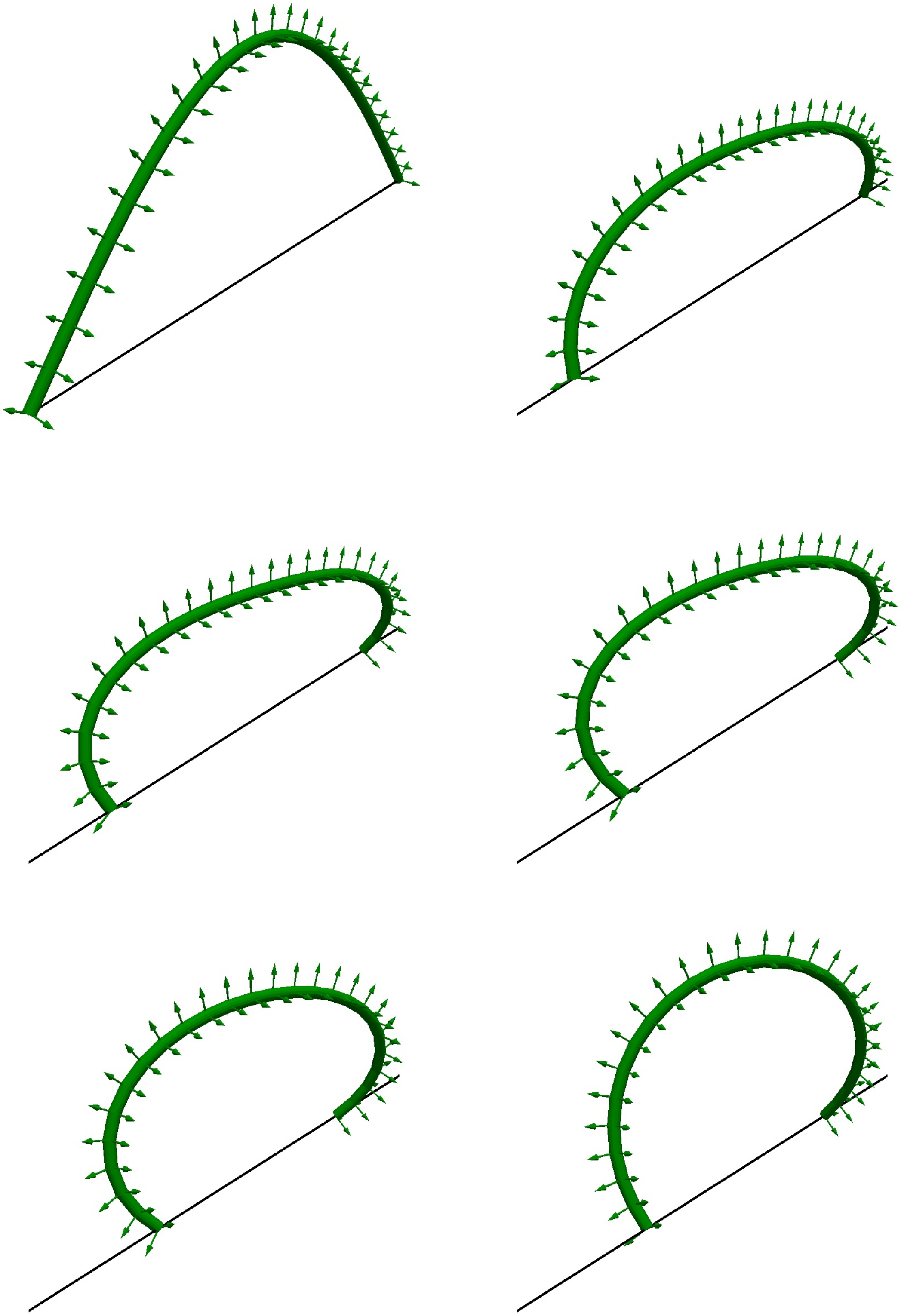


Figure 6.3.1: Snapshots of beam state during solution process. The top left frame is the initial condition, and bottom right frame is the solved state. The black line is the reference configuration of the beam centerline. The arrows represent the directors with an exaggerated magnitude, but the mesh is properly scaled to represent the material surface.

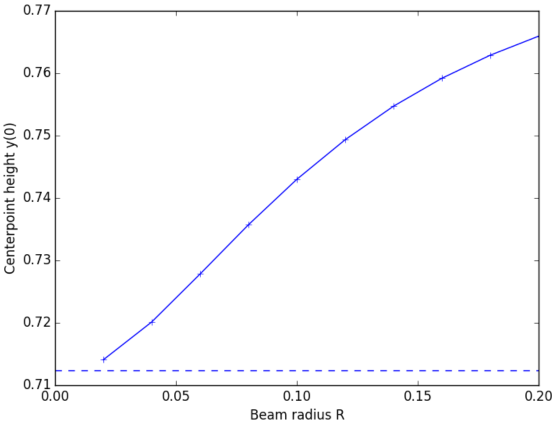


Figure 6.3.2: Decreasing the beam radius to approach thin-string limit with a numerical solution $p = 2$ and $N = 40$. The analytical solution is shown by the horizontal dashed line.

the radius is shown in Figure 6.3.2. Forty quadratic elements were used in each simulation, which, from the study below, was sufficiently accurate. A radius any smaller than $r = 0.02$ became too difficult to solve, requiring more time steps and computational effort than would be desired with which to perform a convergence study. The numerical and analytical solution are in agreement within the limits of their differing assumptions. The analytical solution itself requires a numerical solution step to solve the implicit equation to determine ω, providing a source of numerical error of the “true” solution.

To verify the model, a convergence study for this problem is performed, varying the polynomial order and discretization size. Linear, quadratic, cubic, and quartic elements were used. The same polynomial order for the centroid position and directors is used. Note that the finite element basis only effects the order of the solution along the axis of the beam; the cross sections remain linear. The results are shown in Figure 6.3.3. Since the problem solved analytically and the problem solved by the finite element model are not exactly the same, a “best” solution must be used instead of the analytical solution to estimate converge orders. The most refined and highest order numerical solution is taken to be the “best” solution; in this case it was a quartic, $p = 4$, polynomial order with $N = 35$ elements. The error is measured by comparing the center point displacements,

$$e = \left| y^N (0) - y^{Best} (0) \right| .$$

Since the solutions are being probed in the interpolation space, a convergence order of $p+1$ is expected: i.e., $e = \mathcal{O}\left(h^{p+1}\right)$. The convergence orders were estimated using the linear regression module in SciPy[66] using the data in Figure 6.3.3. The obtain orders are shown in Table 6.2. The obtained orders are in line with expectations.

6.4 Problem 2: Convergence of Dynamic Analysis

The thermal and electrical problems are now considered as well in a dynamic analysis. In a similar set up to before, the beam is pinned at both ends and an applied voltage

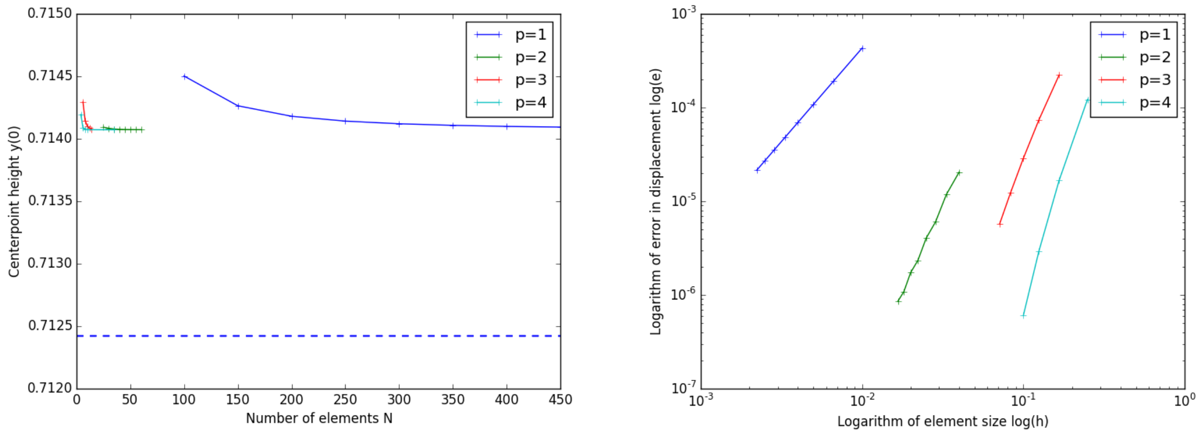


Figure 6.3.3: Convergence of the beam solution with decreasing element size, $h = 2L/N$, and increasing polynomial order p. Left, displacement of center point against number of elements, with analytical solution represented by the horizontal dashed line. Right, log-log plot error as element size decreases, with error measured with respect to the solution at $p = 4$ $N = 35$.

Table 6.2: Convergence orders

Polynomial Basis	Convergence Order
1	1.99387087127
2	3.65015679928
3	3.99825393249
4	5.75578992219

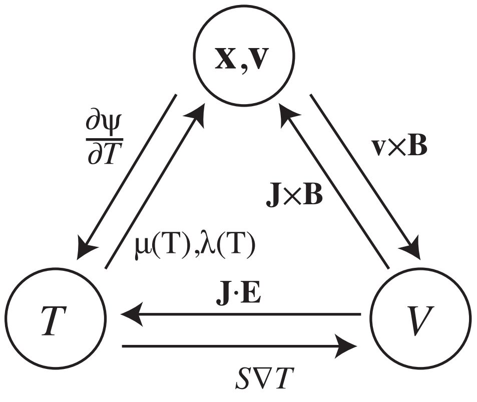

$\mathbf{x},\mathbf{v}$
$\frac{\partial\psi}{\partial T}$
$\mathbf{v}\times\mathbf{B}$
$\mathbf{J}\times\mathbf{B}$
μ(T),λ(T)
$\mathbf{J}\cdot\mathbf{E}$
T
V
$S\nabla T$

Figure 6.4.1: Coupling diagram of fields.

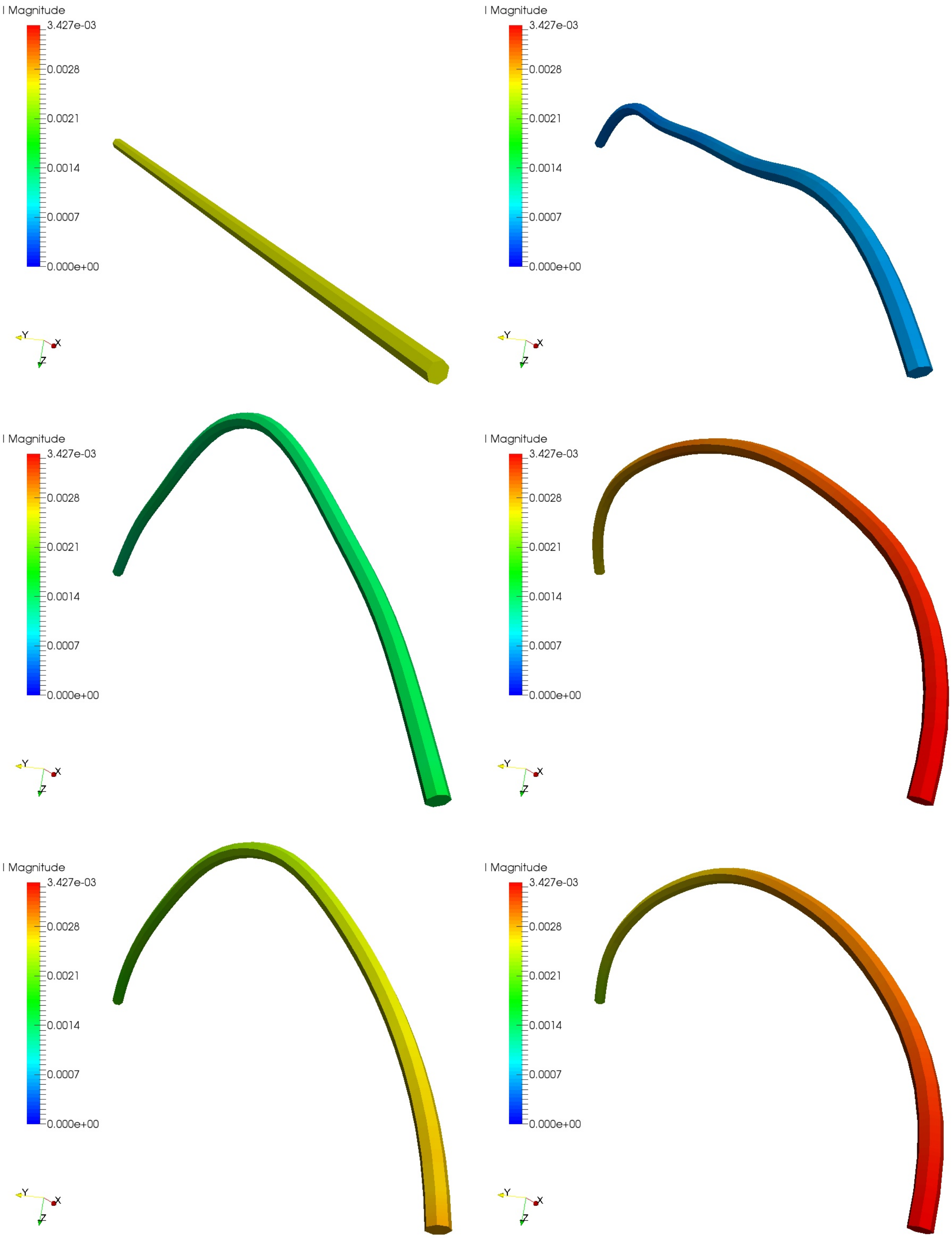


Figure 6.4.2: Time series of the model problem with frames spaced equally apart. The mesh is colored by the current magnitude, measured in amps, with the same ranging in each frame. (The current magnitude is not exactly uniform along the beam at each point in time, though it may not be discernible from the color range.)

Table 6.3: Dynamic problem parameters

ΔV	$\frac{2}{5}V$	Δ	$0mm$
B	$1T$	r	$0.02mm$
ϕ	$\frac{\pi}{4}$	γ	$0.01\frac{Ns}{m}$
E	$10MPa$	σ	$10\frac{S}{mm}$
α_μ	-0.1$\frac{MPa}{K}$	S	$0.1\frac{V}{K}$

and external magnetic field will cause it to deform. The three fields are fully coupled; the effects by which they effect one another is illustrated in Figure 6.4.1. The temperature-to-mechanical coupling is added using a temperature dependent constitutive law for the first Lamé parameter,

$$\mu\left(T\right) = \mu_0 + \alpha_\mu T$$

which is used in the Neohookean strain energy law. The Seebeck effect is used to add a temperature-to-voltage coupling, as described in Section 2.6. The effect modifies the equation for the electric potential to

$$\nabla_x \cdot \epsilon_r \nabla_x V = \nabla_x \cdot \epsilon_r \mathbf{v} \times \mathbf{B} - \nabla_x \cdot \epsilon_r S \nabla_x T,$$

adding another source term based on the gradient of the temperature. Without these two constitutive responses, the temperature field would not be required to solve the system and could be post-processed. The clamps remain fixed, and the electric current varies due to the motion of the beam inducing a back electromotive force. The parameters used for this simulation are shown in Table 6.3. In addition to the pinned boundary conditions on the centroids, at both of the end points the temperature change is held fixed at $0K$ and the voltage is held fixed at $\pm\frac{1}{5}V$. (This is not absolute zero, of course; the temperature is normalized so that the initial temperature of the material is 0.)

The beam is discretized in space using forty elements and all of fields have linear basis functions. Since there is no even approximate analytical solution for comparison, the spatial discretization is not important and is kept low to decrease the computational cost of each time step. Four locations are probed in the solution:

1. Vertical displacement at the center, $y\left(l = 0\right)$
2. Lateral displacement a quarter of the way through the beam, $z\left(l = {}^{1}\!/_{2}\right)$
3. Temperature a quarter of the way through the beam, $T\left(l = {}^{1}\!/_{2}\right)$
4. Voltage a quarter of the way through the beam, $V\left(l = {}^{1}\!/_{2}\right)$

Only the first probe is placed at the center because the (anti-)symmetry of solution causes uninteresting behavior in the other fields at that point. The values of these fields at the final time t_{max} are used to determine the accuracy of these schemes. Since there is no available analytical solution for this problem, even an approximate one as was the case previously, the solutions are compared to an over-refined case of three stages and one thousand time steps, $s = 3$, $N_T = 1,500$. The deformation and current field of the beam is rendered in Figure 6.4.2 and the fields probed are plotted in time in Figure 6.4.3. It should be noted that the current used to color Figure 6.4.3 is not a discretized field: it is post-processed from the other fields. The primary field that is actually solved for the electrical problem is the voltage.

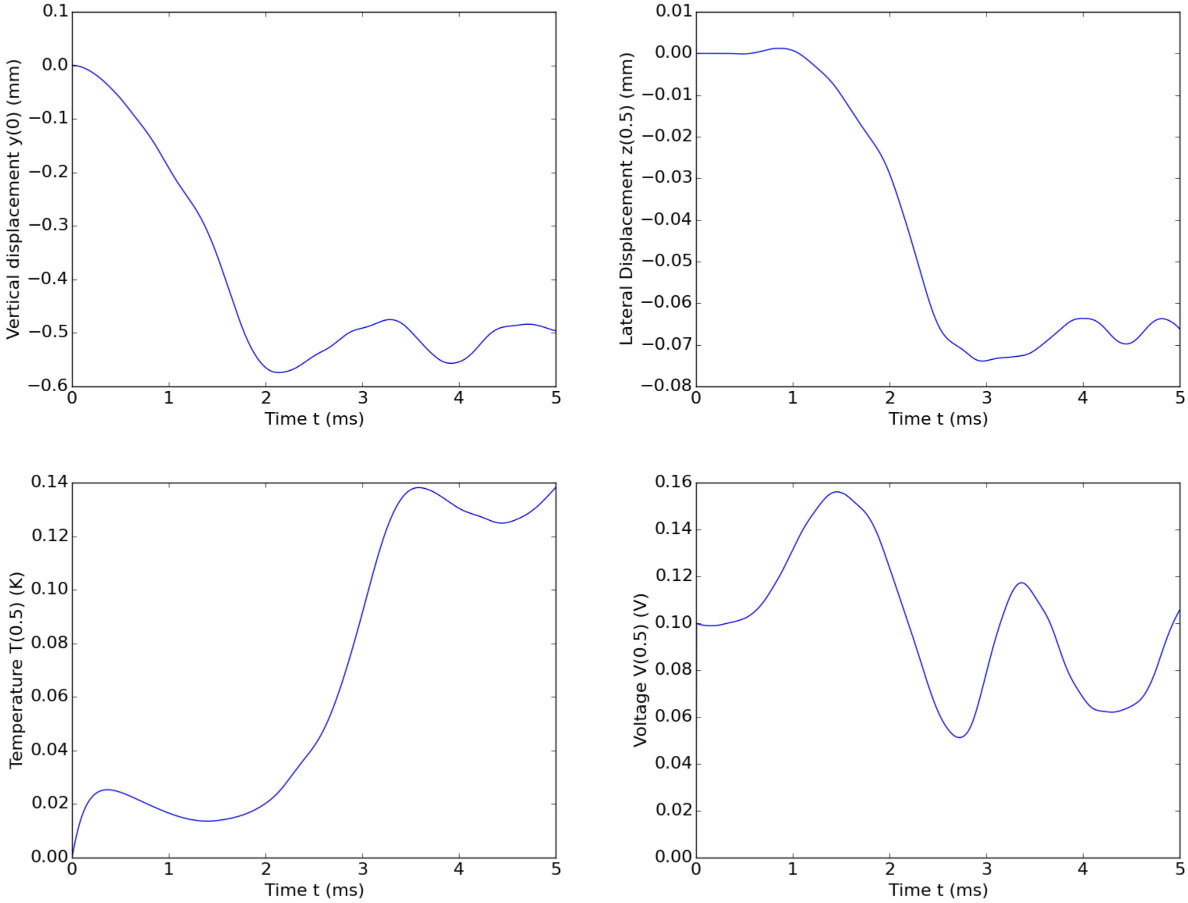


Figure 6.4.3: Time plots of the four probes from the most refined case, $s = 3$, $N_T = 1,000$.

Since the beam is initially at rest, the initial voltage at $l = \frac{1}{2}$ (recall that the beam is of length 2) can be verified with a quick back-of-the-envelop voltage-divider calculation to be equal to $\Delta V \times \frac{R_1}{R_{Total}} - V_{GND} = \frac{2}{5} \times \frac{3R}{4R} - \frac{1}{5} = 0.1V$. As the beam slows down, it approaches this value again, though the simulation is specifically stopped before steady state is reached. The voltage along the beam varies as the beam moves through the magnetic field. The temperature field increases unsteadily from the Joule heating term. The strain-energy term and heat flow through the boundary conditions allow the temperature to drop at certain points in time. The large variations in the temperature are due to the change in current as the beam's velocity changes. The beam heats up initially when the current is high. As it deforms, the current drops from the back electromotive force, and the cooling rate overpowers the joule heating rate. The beam heats up again when the beam slows down and the current increases again, and then cools off when the current drops again during the second large oscillation.

The three L-stable DIRK schemes used are described in detail in Appendix C. The one-stage scheme of the family is Backward Euler. The expected convergence rates for a standard ordinary differential equation for these schemes are $\mathcal{O}\left(h^s\right)$, where s is the number of stages ($s = 1, 2, 3$). The errors for each of the fields as each of the DIRK methods are refined

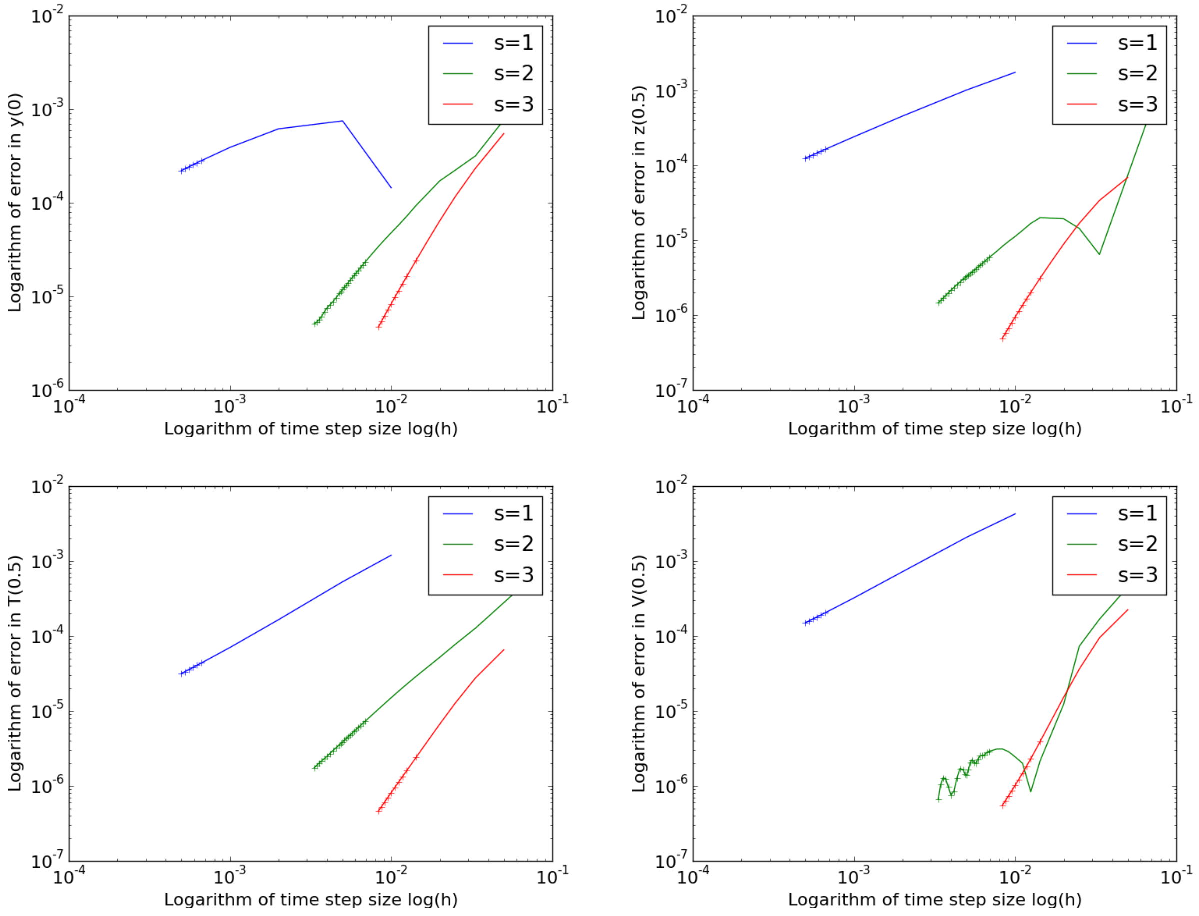


Figure 6.4.4: Error analysis for the three DIRK methods. Each time step size executed is shown; the final "converged" points used for the order analysis are marked with crosses.

in time are shown in Figure 6.4.4. The convergence orders were computed with the points marked by crosses, and the results are tabulated in Table 6.4. The error in the Voltage field for the second order scheme is not monotonic at the end of the simulations sampled, but is decreasing at about the expected rate. The monotonic decrease of the other fields for this scheme does provide confidence that the method is working. The fact that the scheme behaves differently for only this field compared to the first and third order method merits further investigation. The error tolerance for the Newton's method iterations is set to 10^{-12}, and soS absolute errors near 10^{-8} after 1,000 time steps are not very reliable for convergence analysis due to the accumulation of truncation errors in the solution of each time step. This limits the ability to continue to refine the time step size.

6.5 Conclusion

When using a time stepping scheme, in either a time-dependent problem or a dynamic-relaxation context, each Newton iteration has quadratic convergence as long as the step size

Table 6.4: Computed convergence orders for three considered DIRK time stepping schemes. The expected convergence order is equal to the total number of stages for each scheme.

Field	1-stage	2-stage	3-stage
$y\,(0)$	0.85632857532	2.12478679641	3.04844968836
$z\left(\frac{1}{2}\right)$	0.954843792689	1.89428967169	3.43166533069
$T\left(\frac{1}{2}\right)$	1.1294729926	1.96219295075	3.06047011038
$V\left(\frac{1}{2}\right)$	1.08130305608	1.78783635127	3.61152272495

is sufficiently small. The DIRK methods with a monolithic solution are able to solve the coupled differential algebraic equations. The methods converge as expected for standard ordinary differential equations in all of the fields, including the implicit voltage field. Future work on this topic should compare the performance of decoupled iterative solution and explicit/implicit splitting time stepping methods to the monolithic methodology. Other families of Runge-Kutta methods that do not have the L-stable property should also be tried.

The spatial discretization was unchanged when analyzing the dynamic coupled problem. Further study is merited towards analyzing the effect of using mixed-order basis functions for each of the fields, e.g. using quadratic elements for the mechanical fields and linear elements for the temperature and voltage fields. However, it was observed that there was no restriction in using the same basis functions for each of the fields.

Chapter 7

Material Prediction through Static Analysis

7.1 Introduction

Even with the simplified beam-based model developed, the number of degrees of freedom required to simulate a large-scale textile with the complete discretization is intractable. Representative volume elements can be used to predict material properties quickly and efficiently to inform the design of new fabrics.

Simple fabrics without a matrix material are considered. Individual fibrils are free to slide past one another, producing a plastic-type behavior at the macro-scale. The contact laws can take into account interstitial media that effect surface resistances and friction. Plastic deformation and failure of the individual fibrils is not considered, but it is noted that the constitutive responses could easily be extended. Since a static analysis is being considered, sliding-friction models have no bearing. Static-frictional contacts are not considered (and would require more effort to implement.) The framework developed can handle arbitrary microstructures.

A simple four-yarn-by-four-yarn woven fabric unit cell is considered in this chapter as an example. Four yarns are used in each direction to negate the effects of boundary conditions. The yarns are laid out in two directions, called the warp and weft direction. Due to the manufacture process, wherein the warp yarns are held taught and the weft yarns are interwoven through them, the weft yarns are typically thinner and lighter than the warp yarns.

7.2 Homogenization

The microstructure creates an inherently anisotropic unit cell for a continuum model. The deformation of the fibrils at the microscale creates cross-couplings between the mechanical fields and the thermal and electrical fields, so that the bulk conductivities and permittivities will be strain dependent. A continuum model could also be developed for the electric responses, but this would be too broad to model patterns of insulating fibrils and conducting fibrils required for some applications, such as complex electronic circuits. Another strategy is to model each unit cell as an electrical network with property-dependent resistors, with

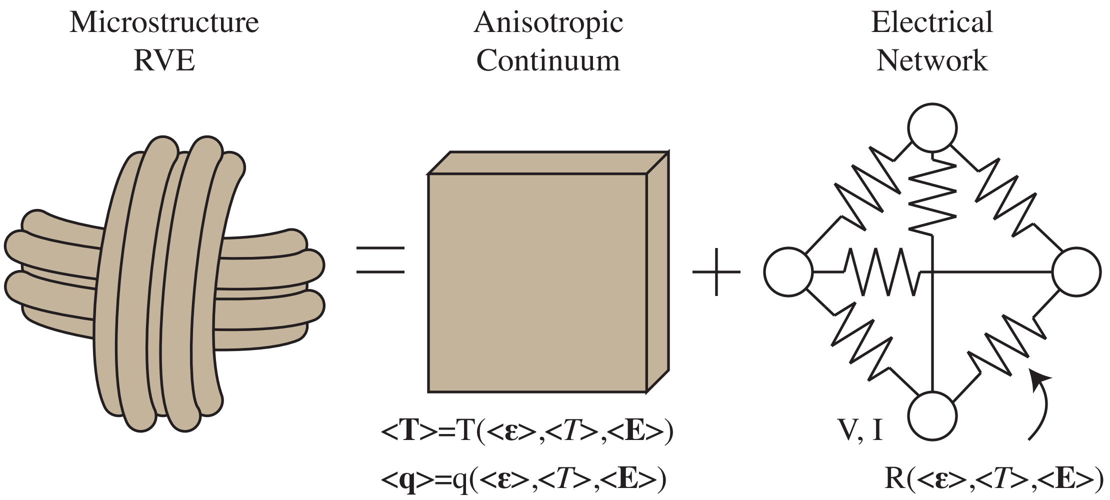


Figure 7.2.1: Homogenization of a microstructure into an anisotropic solid and an electrical network

a one-to-one mapping between unit cells and network cells. A network model can be more readily incorporated into standard electric circuit analysis methods. For example, see [111] for development of resistor network model from experimental data. The homogenization process is illustrated in Figure 7.2.1.

At the microscale level, the stress, displacement, heat flux, and thermal fields are locally variable. To create a macroscale model, it is desired to average out the microscale fields to create constitutive responses for the average stress $\langle \mathbf{P} \rangle$ based on the average strain, $\langle \mathbf{E} \rangle$,

$$\langle \mathbf{P} \rangle \quad = \quad \hat{P}\left(\langle \mathbf{E} \rangle\right)$$

where the average of a tensor is defined as $\langle \mathbf{A} \rangle = \frac{1}{|\Omega|} \int_{\Omega} \mathbf{A} d\Omega$. Producing an expression for $\hat{P}$—e.g. an empirical response of the form $\langle \mathbf{P} \rangle = a_1 \langle \mathbf{E} \rangle + a_2 \langle \mathbf{E} \rangle^2 + ...$—can be challenge. The simulation will be used to tabulate responses instead that can be used in a look-up-and-interpolate material model. Such tables can be implemented in a finite element model of a macroscale textile easily. The desired quantity at the macroscale is the divergence of the stress, $\nabla_X \cdot \mathbf{P}$, which yields an averaged traction,

$$\langle \nabla_X \cdot \mathbf{P} \rangle = \int_{\Omega_0} \nabla_X \cdot \mathbf{P} d\Omega_0 = \oint_{\partial\Omega_0} \mathbf{P}\mathbf{N} d\Gamma_0$$

can be determined by integrating along the boundary of the unit cell. Because the textile unit cell is made up of discrete bodies, the average traction on a boundary is calculated by a summation,

$$\langle \mathbf{t} \rangle = \frac{1}{|\Gamma_0|} \sum \int_{\Gamma_0} \mathbf{P}\mathbf{N} d\Gamma$$

which is related to the total force on the boundary by $\mathbf{f} = |\Gamma_0| \langle \mathbf{t} \rangle$. This will be used to

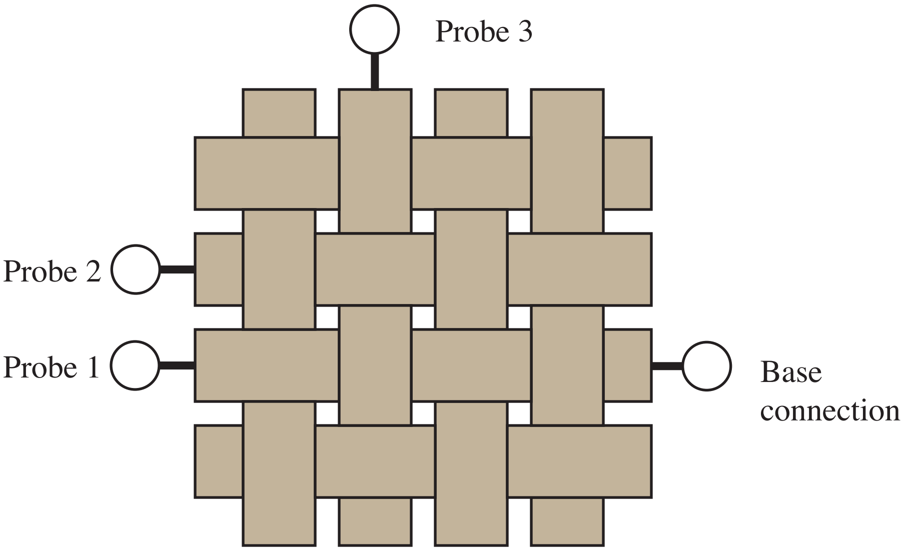


Figure 7.2.2: Probe locations

produce an effective constitutive response for the unit cell by by producing two sets of data

$$\begin{aligned}\mathbf{f}_{warp} &= \hat{f}_{warp}\left(\langle\mathbf{E}\rangle\right) = \sum_{M\cap\Gamma_{warp}}\int_{\Gamma_{warp}}\mathbf{P}\cdot\mathbf{E}_1 d\Gamma \\ \mathbf{f}_{weft} &= \hat{f}_{weft}\left(\langle\mathbf{E}\rangle\right) = \sum_{M\cap\Gamma_{warp}}\int_{\Gamma_{weft}}\mathbf{P}\cdot\mathbf{E}_2 d\Gamma\end{aligned}$$

for each side of the unit cell. The same process can be performed on the heat flux to produce a constitutive response for the heat flux $\langle\mathbf{q}\rangle = \hat{q}\left(\langle\nabla T\rangle\right)$.

Creating an electrical network model does not require any averaging operations. It is required to obtain the total current that passes through the unit cell in response to an applied voltage difference. The applied voltage is the same on the macroscale and microscale. The total current is obtained by integrating the current density along the boundaries of the domain corresponding to the probe locations,

$$I = \int_{\Gamma_1}\boldsymbol{\mathcal{J}}\cdot\mathbf{n}d\Gamma = -\int_{\Gamma_2}\boldsymbol{\mathcal{J}}\cdot\mathbf{n}d\Gamma$$

which should be equal-and opposite if a simple current path is considered. The effective resistance is $\frac{V}{I} = R$. For the textile unit cell, the effective resistance will depend on the strain as well, so the model will be used to produce response data of the form

$$I = \hat{I}\left(V, \langle\mathbf{E}\rangle\right).$$

The resistances of three electrical paths are determined: (1) straight through the same yarn, (2) between two parallel yarns, and (3) two perpendicular yarns. The probe locations are illustrated in Figure 7.2.2.

Algorithm 7.1 Sampling procedure using dynamic relaxation

```
1. Find the next equilibrium position for sampling:
   (a) Set the velocity field to a given strain rate.
   (b) Solve the dynamic problem—only solving for the mechanical fields—for $t_{step}$ time.
   (c) Set the velocity field to zero.
   (d) Solve the dynamic problem with a large dampening parameter for $t_{relax}$ time.
2. Record reaction forces and electric currents:
   (a) Set the velocity field to zero.
   (b) Integrate $\mathbf{f} = \sum_I \int_{\Gamma_I} \mathbf{P} \cdot \mathbf{E}_3 d\Gamma$ to determine the total reaction force on desired
       boundaries.
   (c) For each set of probe locations:
       i. Set the voltage field to match the desired probe values
       ii. Solve the electric potential problem with the relaxed mechanical fields with
           applied potential boundary conditions
       iii. Integrate $I = \sum_I \int_{\Gamma_I} \boldsymbol{\mathcal{J}} \cdot \mathbf{e}_y d\Gamma$ to determine the total current on desired bound-
            aries.
       iv. Solve the thermal problem
       v. Save fields to a file
3. Repeat until all desired samples have been taken.
```

7.3 Methodology

States of the material after heavily dampened relaxation times are taken to be the equilibrium states of the material. The sampling procedure is detailed in Algorithm 7.1. This system only works assuming a one-way coupling between the mechanical state of the material and the thermal and electric fields. If more complicated constitutive laws are considered, the thermal and electric problems need to be varied independently, so that many more samples of the material state would be needed. E.g. the total number of samples would be $N_{mech} \times N_{ther} \times N_{elec}$ for the number of states for each field. The state of the fabric is saved after every stretching iteration as a precaution against computer crashes and power-outages and to enable further analysis after program termination by loading each of the configurations and calculating new resistances and heat transfer coefficients.

In the mechanical analysis, the centroids are pinned at their initial condition. The cross sections are free to pivot to allow the yarns to reorient so that there is a no-applied-torque boundary condition on the directors. The boundary conditions can be written for the strong-form problem as

$$\begin{aligned} \Delta \mathbf{vr} &= \mathbf{0} \,\text{on}\, \partial\Omega \\ \int_A \xi_1 \bar{\mathbf{t}} d\Gamma &= \mathbf{0} \,\text{on}\, \partial\Omega \\ \int_A \xi_2 \bar{\mathbf{t}} d\Gamma &= \mathbf{0} \,\text{on}\, \partial\Omega \end{aligned}$$

where $\Delta\mathbf{vr}$ is the Newton update on the velocity. In the matrix system, it is Newton updates that are solved, so specifying that the velocity at the boundaries should not change enforces

Table 7.1: Simulation Parameters for plain woven fabric

W	$8mm$	σ	$10\frac{S}{mm}$
ρ	$1.0\frac{g}{mm^3}$	k	$1.0\frac{W}{mmK}$
E	$1.26MPa$	h_C	$0.1\frac{W}{mm^2K}$
ν	0	σ_C	$1.0\frac{A}{mm^2V}$
R	$0.15mm$	P^*	$5\frac{N}{mm}$
S	$0\frac{V}{K}$	γ	$0.1\frac{Ns}{m}$
α_μ	$0\frac{MPa}{K}$		

that the velocity remain at the prescribed value. Because the boundary conditions are on the Newton update and the velocity is imposed directly, the same boundary conditions can be used in both the stretching phase and the relaxing phase.

The voltage field has Dirichlet boundary conditions only on the probe locations, Γ_B and Γ_P, where Γ_P matches only one of the three locations illustrated in Figure 7.2.2. Even through the problem for the potential happens to be linear in this case, the same Newton's method solver is used. The desired values for the Dirichlet boundary conditions are set via an initial condition, and the boundary conditions applied to the matrix system are on the Newton update, which must be set to zero. The remaining boundaries are insulating so that $\bar{\mathcal{J}} \cdot \mathbf{n} = 0$. The boundary conditions on the PDE are

$$\begin{aligned} \Delta V &= 0 \,\text{on}\, \Gamma_B \text{and} \Gamma_P \\ -\boldsymbol{\sigma}^{-1}\boldsymbol{\mathcal{J}} \cdot \mathbf{n} = \boldsymbol{\epsilon}_r \nabla_x V \cdot \mathbf{n} &= 0 \,\text{on}\, \partial\Omega \backslash \left(\Gamma_B \cup \Gamma_P\right) \end{aligned}$$

where ΔV is the Newton update on the voltage. The effect of Joule heating in the fabric is to be observed. The temperature is pinned to zero (difference from the reference temperature) on the boundaries. Similarly to the electric potential, the boundary conditions are set with initial conditions and the matrix system boundary condition is a zero update, or

$$\Delta T = 0 \,\text{on}\, \partial\Omega.$$

7.4 Results

In this example, all of the fibrils are the same. The properties are listed in Figure 7.1. The initial configuration and the relaxed state are shown in Figure 7.4.1. The reference configurations of each of the fibrils is straight, and the fibrils would intersect with crossing fibrils if started from the reference configuration. Each yarn is initially in a three-level stack of with a 3-4-3 fibril count pattern (see the bottom left of Figure 7.4.1.) After the relaxation process, it is observed in in the cross section through one of the center yarns shown in Figure 7.4.2 that the yarns flatten out into a two-level stacking.

Two deformations are considered: uniaxial extension and simple shearing. The velocity field used to drive uniaxial extension/compression (Step 1a in Algorithm 7.1) is

$$\mathbf{v} = \frac{\Delta_{step}}{t_{step}} \frac{X_1}{L_0} \mathbf{e}_1.$$

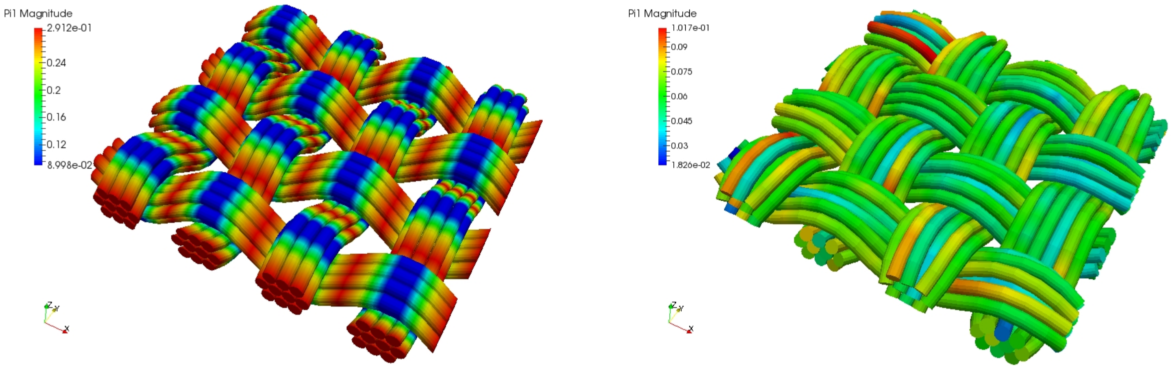


Figure 7.4.1: Initial condition geometry (top) and dynamically relaxed geometry (bottom). The mesh is colored by total reaction force across the cross section, $\int_C \mathbf{PE}_3 dA$, measured in newtons.

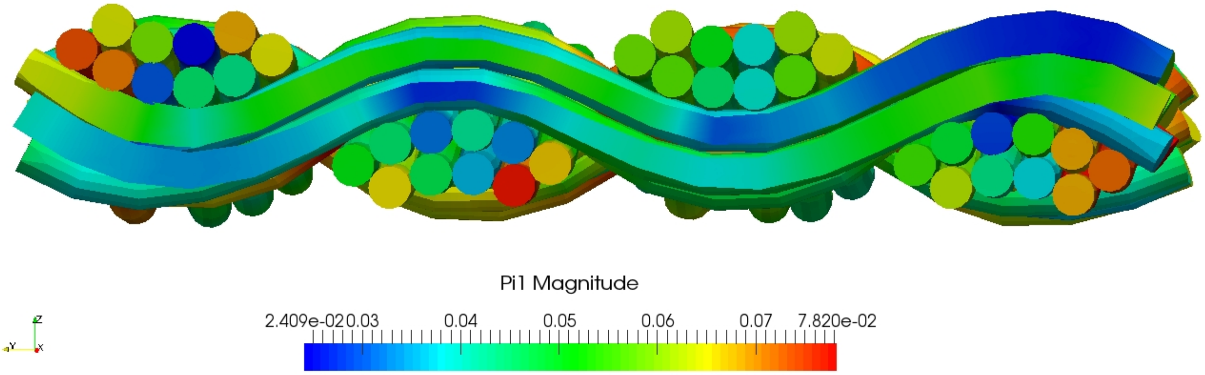


Figure 7.4.2: Cross section of relaxed plain weave. The internal reaction force is measured in newtons.

The velocity field used to drive the simple shearing is

$$\mathbf{v} = \frac{\Delta_{step}}{t_{step}} \frac{X_1}{L_0} \mathbf{e}_2.$$

In these equations, Δ_{step} is the desired strain to be applied during the step over the course of t_{step} time. The unit cell is brought to 25% strain over the course of ten steps for both cases. The final strained states are rendered in Figure 7.4.5. The calculated force and current responses are plotted in Figure 7.4.3 for the uniaxial extension case and in Figure 7.4.4 for the for the shearing case. The z direction corresponds to out-of-plane reactions. Because only in plane motion is considered, the reaction forces stay new zero in this component.

The endpoints of each of the fibrils is held fixed at the reference length. Because of the interweaving, this places each fibril in tension, though the stress fields are not uniform as seen in Figures 7.4.1 and 7.4.2. Some fibrils in the corners have particularly high stresses, such as the one visible at the top of the figure, where they are unable to move due to the pinned boundary conditions. The unit cell is thus originally in tension, with approximately two newtons of force acting on the clamps in both directions. A relaxation process could be applied in which the yarns at the boundary were allowed to move inwards to bring the initial tractions to zero.

The variation in the resulting currents for this static analysis is due to the strain is dependent entirely upon conduction through contacts. Using the constant isotropic conductivity constitutive law, the electrical resistances do not change due to the deformation because of the transformation of the conductivity tensor. The fibrils can be analyzed separately if they are insulated, allowing the electrical problem to always be pushed back into the reference configuration

$$\begin{aligned} I &= \int_{\Gamma} \boldsymbol{\sigma} \nabla_x V \cdot \mathbf{n} d\Gamma \\ &= \int_{\Gamma_0} \left(\frac{1}{J} \mathbf{F} \boldsymbol{\sigma}_0 \mathbf{F}^T \right) \left(\mathbf{F}^{-T} \nabla_X V \right) \cdot J \mathbf{F}^{-T} \mathbf{N} d\Gamma_0 \\ &= \int_{\Gamma_0} \boldsymbol{\sigma}_{\mathbf{0}} \nabla_X V \cdot \mathbf{N} d\Gamma_0 \end{aligned}$$

so that the current passing through the boundary of each fibril would always be same. It is possible to consider constitutive laws for the conductivity tensor that is dependent on the state of the material beyond the transformation from the reference configuration to current configuration. This would allow the resistance to change without contact resistances present.

The current distributions through the fabric in these deformed states are shown in Figure 7.4.6. As can be seen in Figure 7.4.6, an appreciable amount current passes through neighboring yarns in all cases. With the network model illustrated in Figure 7.2.1, the unit cell is contains 16×2 yarn resistors and 16×4 contact resistors. Assuming that there are only two types of resistors that see the same input stress state, $R_y(\langle \mathbf{E} \rangle)$ and $R_c(\langle \mathbf{E} \rangle)$, the three probes could be used to calculate the resistances for the unit cell. The temperature distributions in the unit cell due to the Joule heating effect are shown in Figure 7.4.7.

A larger current corresponds to a lower resistance. As could be expected, the resistance measured along a single yarn, Probe 1, is always the lowest. However, the additional conducting paths present due to the electrical contacts allow the resistance to vary significantly

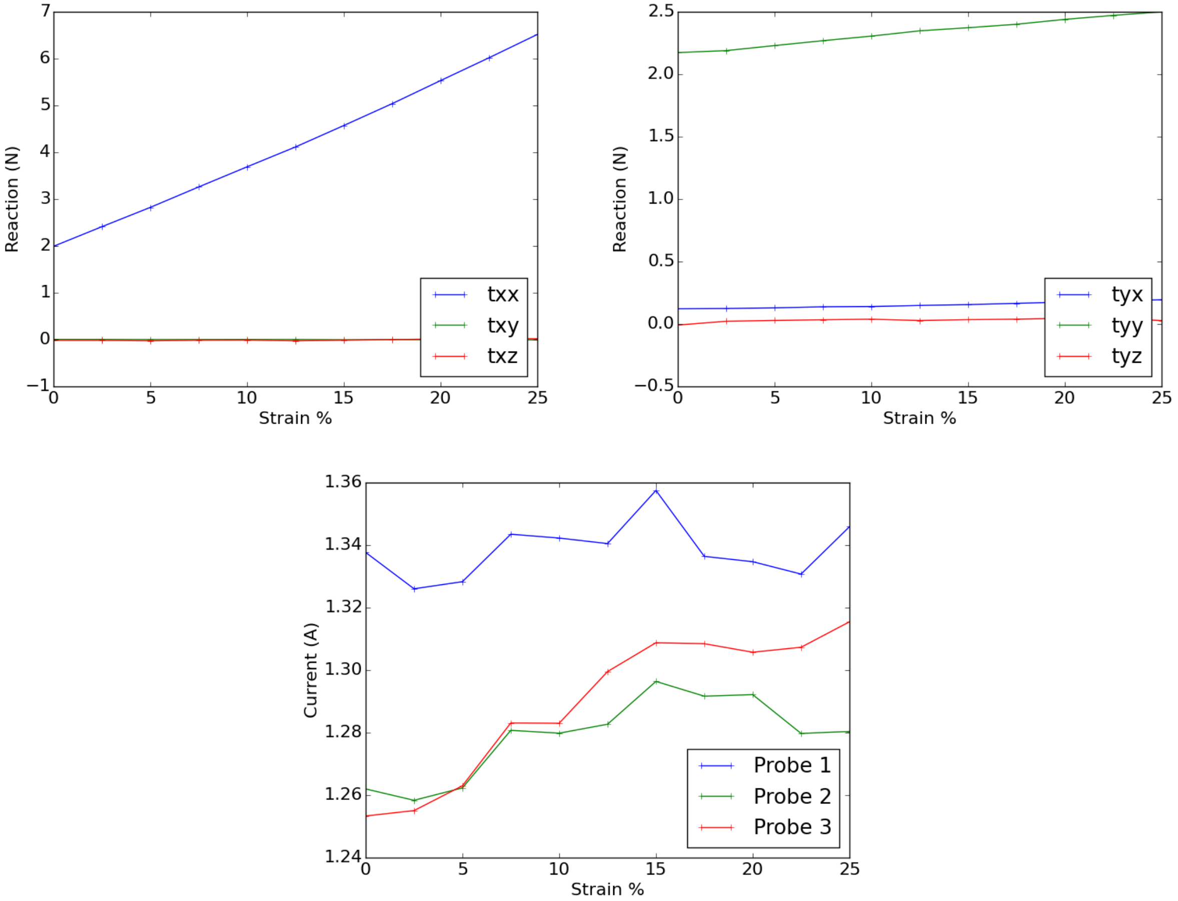


Figure 7.4.3: Reaction forces and currents for the uniaxial stretching case. A test voltage of $1V$ is applied, so the effective electrical resistance is the reciprocal of the measured currents.

as the textile deforms. The contact resistance can change drastically as fibrils slide by each other. This causes the response current to change non-monotonically, and even reverse trend at some points. The resistance can be decreasing due to an increase in contact pressure between two fibrils, but at critical point as the deformation continues the fibrils can rearrange to a state with lower contact pressures resulting in a higher resistance. The increase in contact pressures and rearrangement of contacts cause the resistances for Probes 2 and 3 to overall drop significantly as the textile is stretched in both cases.

7.5 Conclusion

The fibril configurations studied are not necessarily unique. Due to the discrete nature of the structures, other packings are possible that will result in different local stresses that may affect the macroscopic response. A more detailed study would repeat the analyses on perturbed initial states to quantify the range of responses.

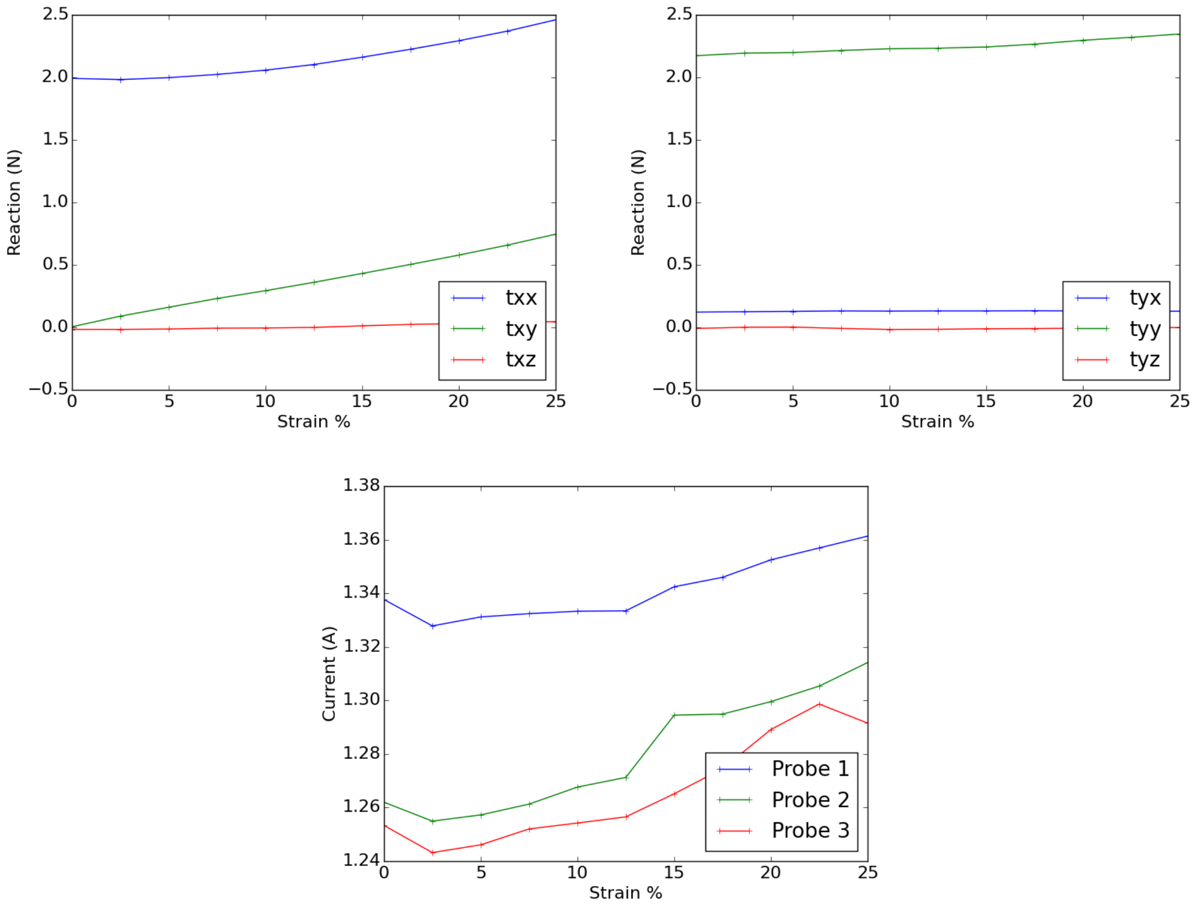


Figure 7.4.4: Reaction forces and currents for the uniaxial shear case.

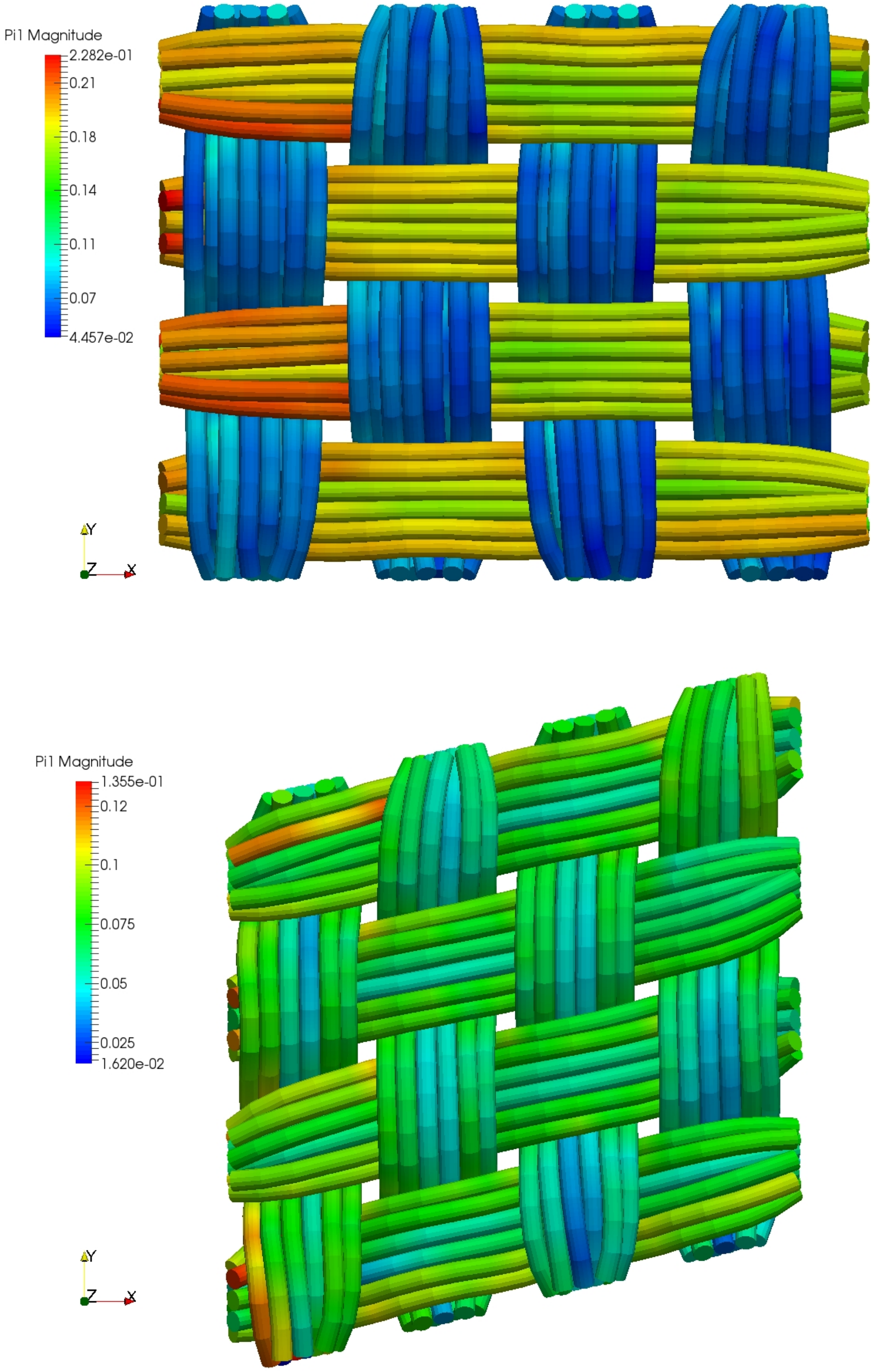


Figure 7.4.5: Final stress distributions at 25% strain for stretching (top) and shearing (bottom). The reaction force is measured in newtons.

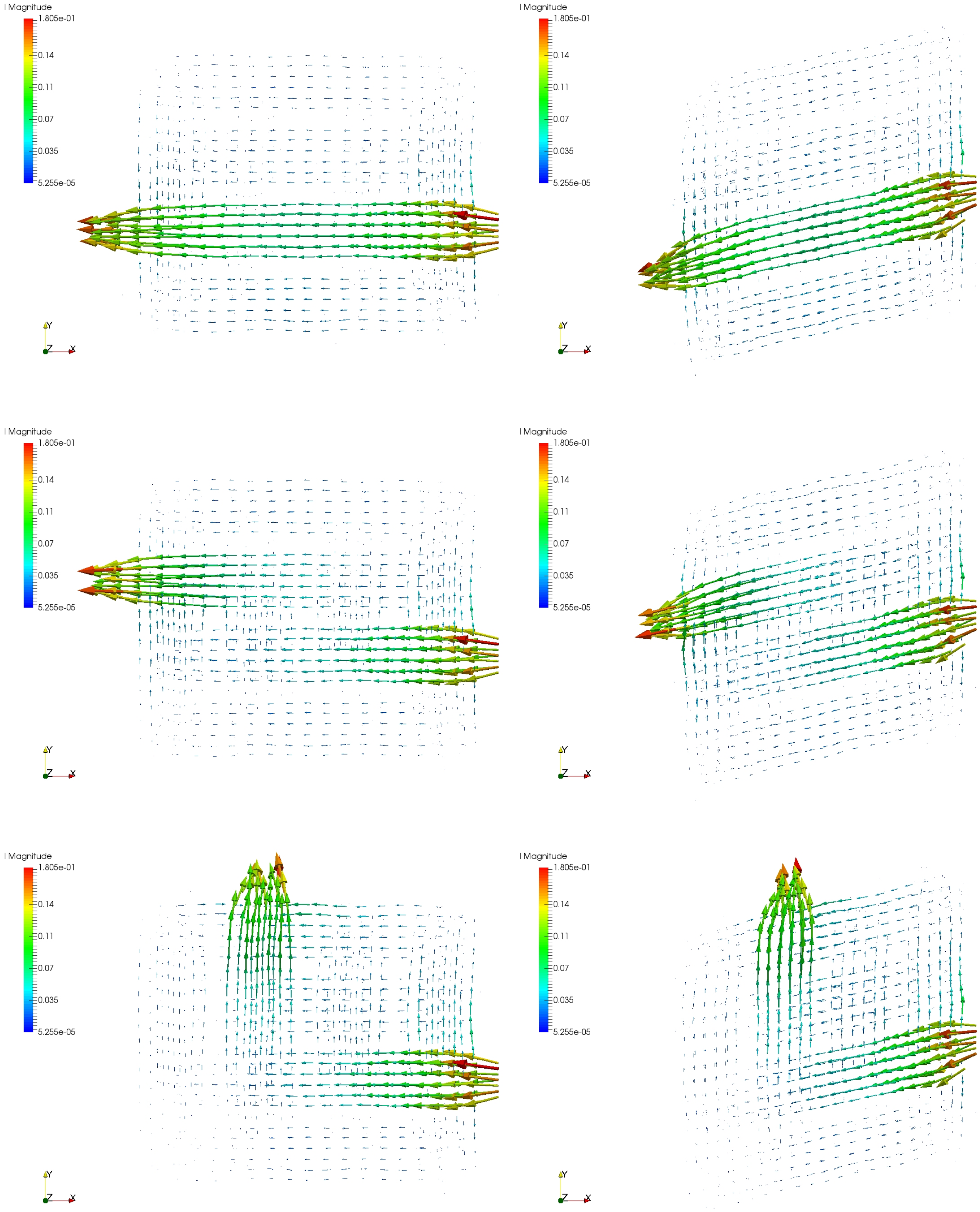


Figure 7.4.6: Current distributions for the three probe locations for the largest calculated strains for the stretching (left) and shearing (right). The current is the total through the cross section, measured in amps.

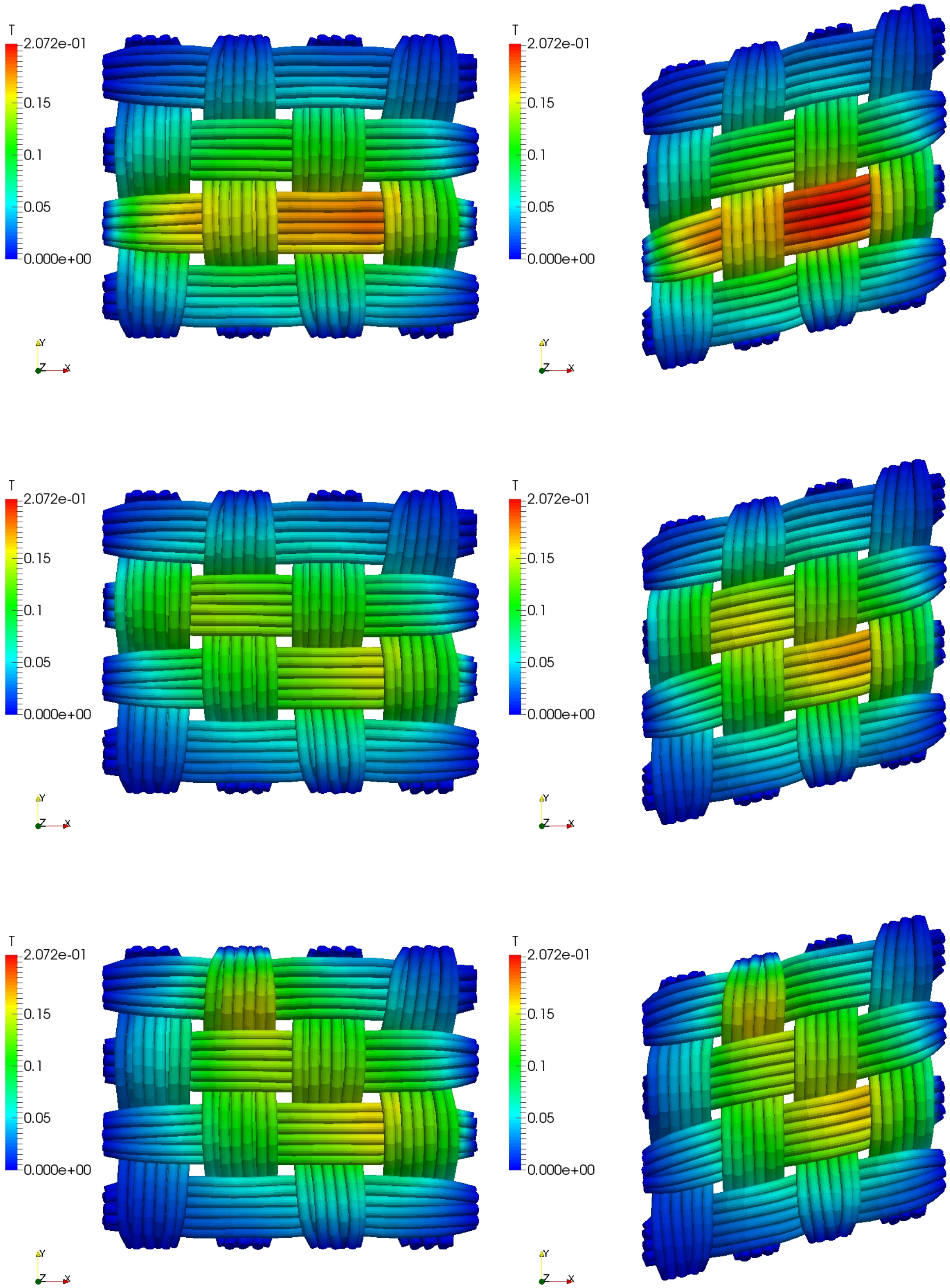


Figure 7.4.7: Temperature distributions due to Joule heating for the three probe locations for the largest calculated strains for the stretching (left) and shearing (right). The temperature is measured in degrees Celsius in relation to the boundary.

Chapter 8

Electromagnetic Armor Simulation through Dynamic Analysis

8.1 Introduction

Many novel armors seek to incorporate electromagnetic elements to enhance their performance. The strategy illustrated in Figure 1.1.2 in Chapter 1 is simulated. It is possible to control the fabric to deflect the direction of the projectile, tipping the projectile in addition to purely stopping the projectile head-on. Turning the projectile to its broad side is desirable because this will increase the contact area, distributing its energy across a large portion and reducing the damage to the remaining armor. An important note is the "one-time-use" design criterion of armors: a successful strategy for negating a threat can be self-destructive. This allows for the exploration of very extreme ranges of electromagnetic fields that would not be normally accessible in standard industrial settings. This allows the capabilities of the computational model to be tested in an extreme scenario where various effects, such as the back-electromotive-force, become significant. Capacitors designed for these applications were demonstrated in [5].

Firstly, the boundary conditions are discussed to address issues in dynamic analyses that do not appear when performing material property prediction. Then, the fully-coupled dynamic problem is employed to observe to observe the resulting deformation of a small textile with different magnetic field orientations. Finally, the electromagnetically enhanced armor is simulated.

8.2 Weakened Voltage Boundary Conditions

It is desired to consider the case where only certain fibrils are connected to an electrical circuit, and the others are insulated. For example, consider the case of a plain-woven fabric where it desired to run current along one direction: both ends of the warp yarns are connected to applied voltages, and the weft yarns are insulted. Trouble arrises in the quasi-static voltage problem when some of the weft yarns are isolated from the warp yarns. If an insulated conductor is not in contact with any body connected to the circuit, the solution to the voltage has an undetermined constant mode on that body. That is, if $V(x)$ is a solution,

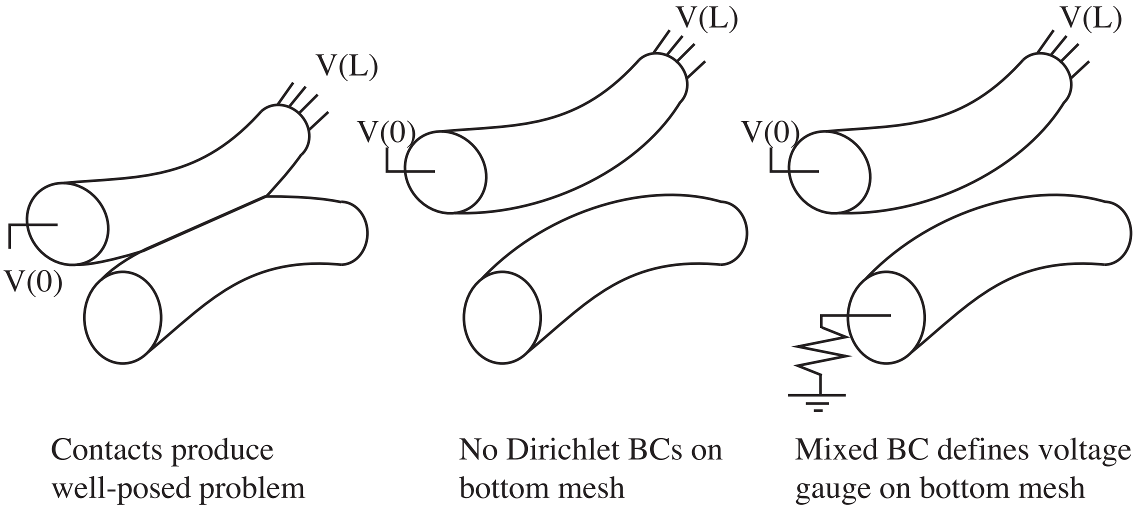

V(L)
V(0)
Contacts produce well-posed problem
V(L)
V(0)
No Dirichlet BCs on bottom mesh
V(L)
V(0)
Mixed BC defines voltage gauge on bottom mesh

Figure 8.2.1: Mixed boundary condition, with a sufficiently large external resistor R, keep the electric potential well defined when insulated conductors are disconnected from the Dirichlet boundary conditions.

then so is $V'(x) = V(x) + C$ for all constants C. For certain configurations of contacts, the problem may be split up into separate groups of fibrils are in contact. If no Dirichlet boundary condition is present in a collection, the sub-problem is ill-posed. There is no unique solution and the final matrix system for the entire set of fibrils will be rank deficient. Depending on the solver type being used, this can be unsolvable. The uBLAS library for sparse linear algebra used would sometimes fail to solve the resulting matrix and other times produce a constant field on unpinned fibrils with a seemingly random value. Both results are undesirable.

To prevent this ill-posed problem from arising, the condition of insulation is weakened on non-connected fibrils. On one end, a mixed boundary condition is substituted of the form given in Equation 2.8.2. As the resistance increases, $R \to \infty$, the boundary condition approaches an insulated one, but when the fibril is isolated, the voltage will be well defined and equal to the external voltage of the boundary condition, ΔV_E. This is analogous to applying pull-down resistors in electronic circuits to keep signals well defined.

If a different electrical discretization was being used in which the interstitial air or vacuum were also discretized in addition to the conducting media, this would not be a concern. The electric potential in the vacuum would couple all of the bodies, even if there were no electrical contacts between the conductors. Of course, there would still need to be a Dirichlet boundary condition somewhere in the system.

8.3 Induced Deformation

The current-carrying yarns on the warp yarns are oriented along x. The angle of the magnetic field is varied to produce three fields: $\mathbf{B}_1 = B\cos\frac{\pi}{4}\mathbf{e}_x + B\sin\frac{\pi}{4}\mathbf{e}_y$, to produce a

Table 8.1: Simulation Parameters

L_0	$12.5mm$	σ	$10\frac{S}{mm}$	B_{mag}	$0.005kT$
L	$10.0mm$	k	$1.0\frac{W}{mmK}$	$\bar{T}$	$0K$
ρ	$0.00144\frac{g}{mm^3}$	h_C	$0.1\frac{W}{mm^2K}$	$\bar{r}_{warp}$	$100\frac{A}{V}$
E	$1.26MPa$	σ_C	$1.0\frac{A}{mm^2V}$	$\bar{J}_{warp}$	$1000\frac{A}{mm^2}$
ν	0	P^*	$500\frac{N}{mm}$	$\bar{r}_{weft}$	$0.001\frac{A}{Vmm^2}$
R_{warp}	$0.4mm$	γ	$0.1\frac{Ns}{m}$	$\bar{J}_{weft}$	$0\frac{A}{mm^2}$
R_{weft}	$0.6mm$	S	$0\frac{V}{K}$	α_μ	$0\frac{MPa}{K}$

cork-screwing effect; $\mathbf{B}_2 = B\mathbf{e}_y$, to produce an upwards force; and $\mathbf{B}_3 = B\cos\frac{\pi}{4}\mathbf{e}_y + B\sin\frac{\pi}{4}\mathbf{e}_z$ to produce an up-and-sideways force. The weft yarns are slightly thicker than the warp yarns, but all other material properties are the same. The simulation parameters are shown in Figure 8.1. Each fibril is meshed with thirty elements using linear basis functions.

The time series of the fabric deformation for the three magnetic fields are shown in Figure 8.3.1. The temperature fields for the final time are shown in Figure 8.3.2. In the first frame, the weft yarns are not in contact with the warp yarns, so their voltage is held near ground by the weakened boundary condition. When the fabric deforms and the yarns come in contact, the voltages of the weft yarns are defined by the contacts.

Because of the initial configuration of the weave, the yarns in the cork-screwing effect, $\mathbf{B}_1$, originally seek a high-circularity configurations. As the fabric tightens, the yarns are relaxed into a lower-circularity configuration, but are still less straight than the resulting deformation for the perpendicular magnetic field. The yarn on the side starts to unravel from the weft yarns by sliding off in a few locations in the $\mathbf{B}_3$ case.

8.4 Electromagnetically Enhanced Armor

Only beam-beam contact methods were implemented, so the projectile is represented as a single director-based element. The elliptical cross section representation has the unintended result that the end points of beams will be hemispherical. That is, if the closest point on one beam is on the end point of the mesh, the equation $R = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$ representing the surface will reduce to round off the endpoints. This effect is exploited to handle projectile–fibril contact.

A 32 yarn by 32 yarn plain woven fabric is modeled. Each yarn is discretized with 80 elements, and the projectile is one element. A mixed boundary condition is applied on one side and a Dirichlet boundary condition on the other so that current only runs through the warp yarns. The weft yarns are insulated from the warp yarns, $\sigma_C = 0$, and are only connected to a weak insulating boundary condition. However, because the voltage PDE is also solved on the weft yarns, there can be local current and charge build up due to nonuniform motion through the magnetic field. The third magnetic field orientation from the previous section is used, $\mathbf{B} = B\cos\frac{\pi}{4}\mathbf{e}_y + B\sin\frac{\pi}{4}\mathbf{e}_z$, so that the projectile is pushed at an angle by the electromagnetic force, both opposing its motion and tipping it. The parameters used in the simulation are shown in Figure 8.2. A relaxation calculation is run first to obtain the resting state of the textile. The projectile is present in this simulation but stationary so

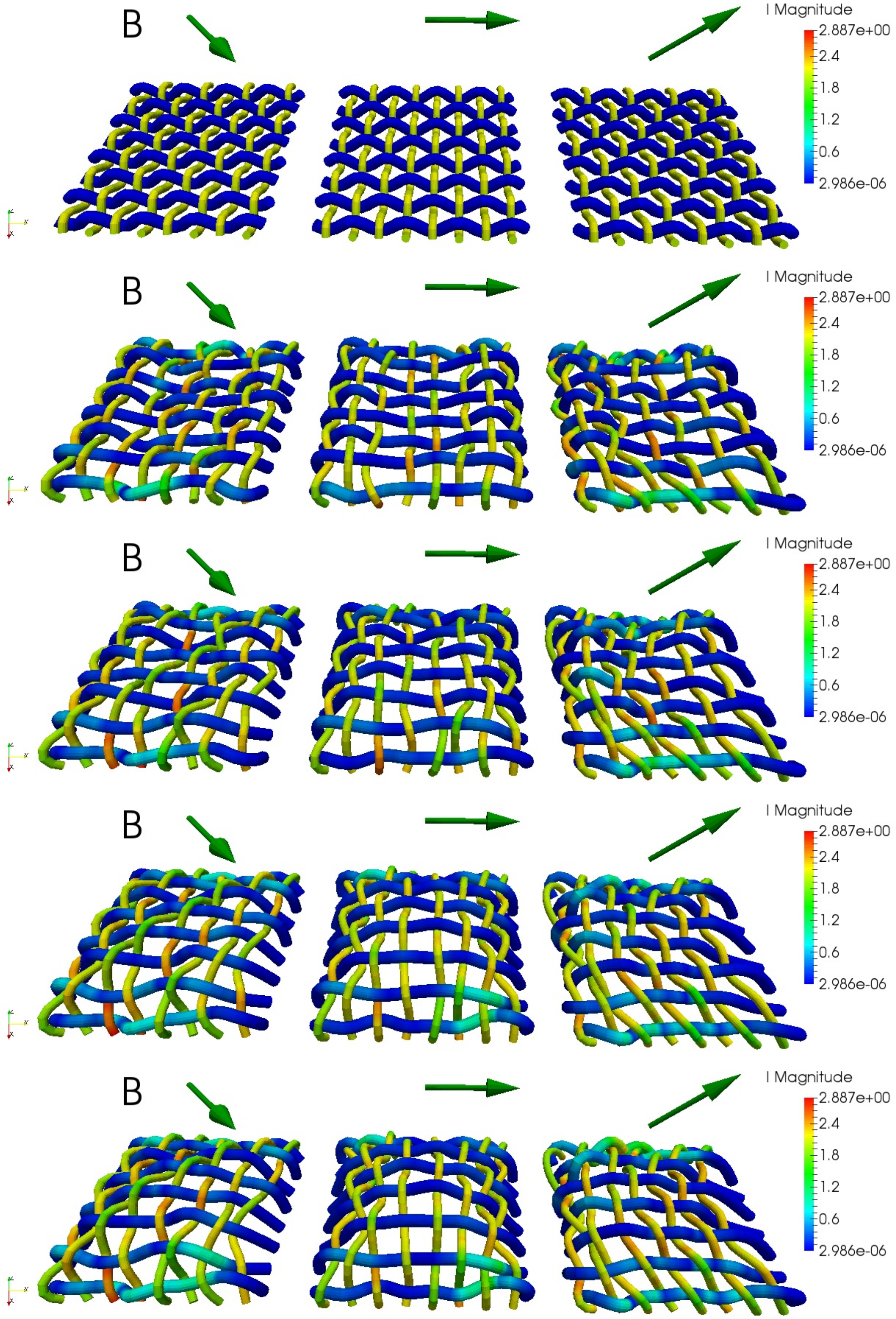


Figure 8.3.1: Time frames of the deformation of three orientations of the magnetic fields, from left to right: $\mathbf{B}_1$ in the xy plane, $\mathbf{B}_2$ along the y axis, and $\mathbf{B}_3$ in the yz plane. The orientation is rendered as an arrow above each. The coloring is by current, measured in amperes.

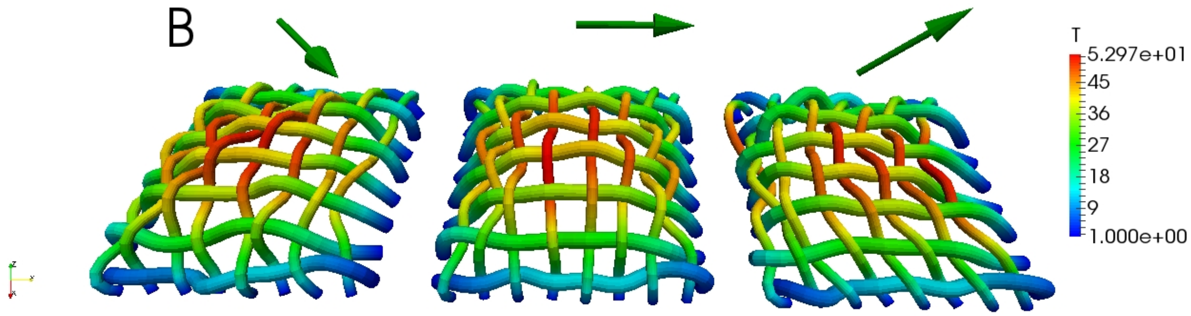


Figure 8.3.2: Final state of the fabrics colored by temperature (degrees Celsius).

Table 8.2: Impact Simulation Parameters

L	$50.8mm$	σ	$10^5\frac{S}{mm}$	B_{mag}	$0.001kT$
ρ	$0.00144\frac{g}{mm^3}$	k	$1.0\frac{W}{mmK}$	$\bar{T}$	$0K$
E	$1.26MPa$	h_C	$0.1\frac{W}{mm^2K}$	$\bar{r}_{warp}$	$100\frac{A}{V}$
ν	0	σ_C	$0.0\frac{A}{mm^2V}$	$\bar{J}_{warp}$	$-100\frac{A}{mm^2}$
R_{warp}	$0.4mm$	P^*	$20\frac{N}{mm}$	$\bar{r}_{weft}$	$0.001\frac{A}{V}$
R_{weft}	$0.4mm$	γ	$0.1\frac{Ns}{m}$	$\bar{J}_{weft}$	$0\frac{A}{mm^2}$
$R_{projectile}$	$18mm$	$\sigma_{projectile}$	$1\frac{S}{mm}$	$\bar{r}_{projectile}$	$0.001\frac{A}{V}$
$L_{projectile}$	$5.5mm + 2R$	$\gamma_{projectile}$	$0\frac{Ns}{mm}$	$\bar{J}_{projectile}$	$0\frac{A}{mm^2}$
$\rho_{projectile}$	$0.01\frac{g}{mm^3}$	S	$0\frac{V}{K}$	α_μ	$0\frac{MPa}{K}$

that the data arrays will have the same size. The file is saved and then loaded as the initial condition for each the impact simulation. After loading the relaxed state, the velocity of the projectile is set to the desired value.

The two-stage second-order DIRK is used. Although an L-stable integration scheme is used, the time step size is strongly limited by the ability to solve the nonlinear system of equations for each stage. The major limiting factor in the regularity of the system is the non-smooth contact laws. The time step must thus be kept small enough to catch new contacts within an update. A useful heuristic for the maximum allowable time step is

$$\Delta t_{max} \approx \frac{1}{20}\frac{R_{min}}{v_{max}}$$

where 20 is an arbitrary "magic" number, dictating that the fasting moving object in the system should take twenty time steps to cross the smallest fibril radius. The maximum velocity can be approximated by the initial speed of the projectile for impact simulations, though it is possible for small pieces of the armor to be moving faster from momentum-transfer at some time. The simulation is performed for $0.5ms$ using 1,000 time steps so that $\Delta t = 0.0005ms$. The resulting deformation is shown in Figure 8.4.1.

The projectile is not rigid, so although it does not have many degrees of freedom, it is able to deform. The end points of the element oscillate, producing an interesting effect. In Figure 8.4.2, the velocity of both ends of the projectile is plotted. This is not necessarily a numerical artifact. If the material properties of the projectile are representative of the

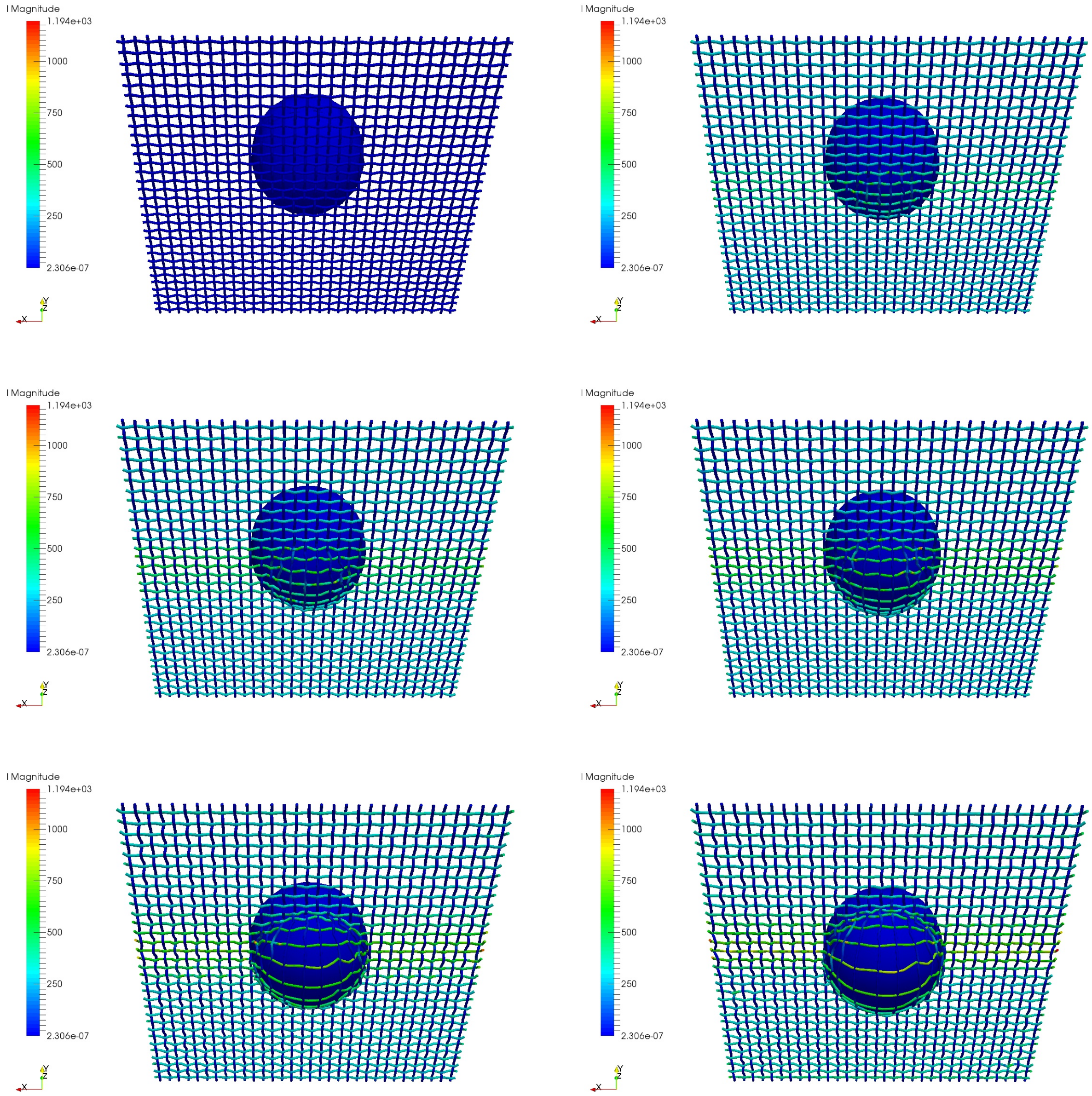


Figure 8.4.1: Equally spaced snapshots of the impact simulation. The coloring is by current, measured in amperes.

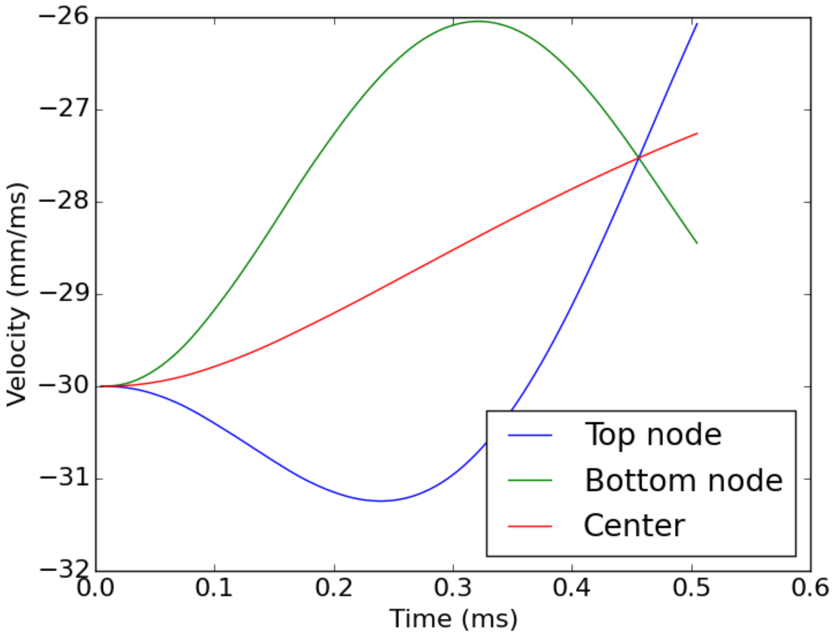


Figure 8.4.2: Projectile velocity in z direction, with the initial velocity being $-30\frac{m}{s}$. The velocity of the center of mass as well as the two end points is plotted. The projectile initially compresses, and at the end begins to expand.

threat of interest, the oscillatory behavior is a very low-fidelity calculation of the stress waves propagating through the projectile. The velocity of the center of mass of the projectile shows a steady decrease as it is slowed down by the fabric. The projectile initially compresses on impact. The angular velocity of the projectile can estimated using the formula for a rigid body, $\boldsymbol{\omega} = \frac{\mathbf{r}_{CM} \times \mathbf{v}_{rel}}{\mathbf{r}_{CM} \cdot \mathbf{r}_{CM}}$. Using the velocities and positions of the top node and bottom node of the projectile, the angular velocity is calculated as

$$\boldsymbol{\omega} = \frac{\left(\frac{1}{2}\mathbf{q}_{top} - \frac{1}{2}\mathbf{q}_{bot} + \frac{L}{2}\mathbf{e}_z\right) \times \left(\frac{1}{2}\mathbf{v}_{top} - \frac{1}{2}\mathbf{v}_{bot}\right)}{\left(\frac{1}{2}\mathbf{q}_{top} - \frac{1}{2}\mathbf{q}_{bot} + \frac{L}{2}\mathbf{e}_z\right) \cdot \left(\frac{1}{2}\mathbf{q}_{top} - \frac{1}{2}\mathbf{q}_{bot} + \frac{L}{2}\mathbf{e}_z\right)}.$$

The angular velocity of the projectile around the x axis is plotted in Figure 8.4.3.

Increasing the current through the textile will increase the amount of force that can be applied. However, the device is limited by its ability to destroy itself through melting. A linear increase in current results in a linear increase in the force, but results in a quadratic increase in the the Joule heating term, since $r = \sigma \mathcal{J}^2$. The temperature of the fabric during the impact is shown in Figure 8.4.5. Interestingly, the fabric is hottest in a ring around around the projectile. Because of the heat flux through contacts, the massive projectile acts as a thermal reservoir and cools the pats of the fabric it is in contact with. On the other hand, projectiles are usually a high temperature, which would increase the rate of heating at the impact site at first.

A close up of the impact site is shown in Figure 8.4.6. The deformation of the yarns due to the magnetic force is visible, where the current carrying warp yarns are being lifted off of

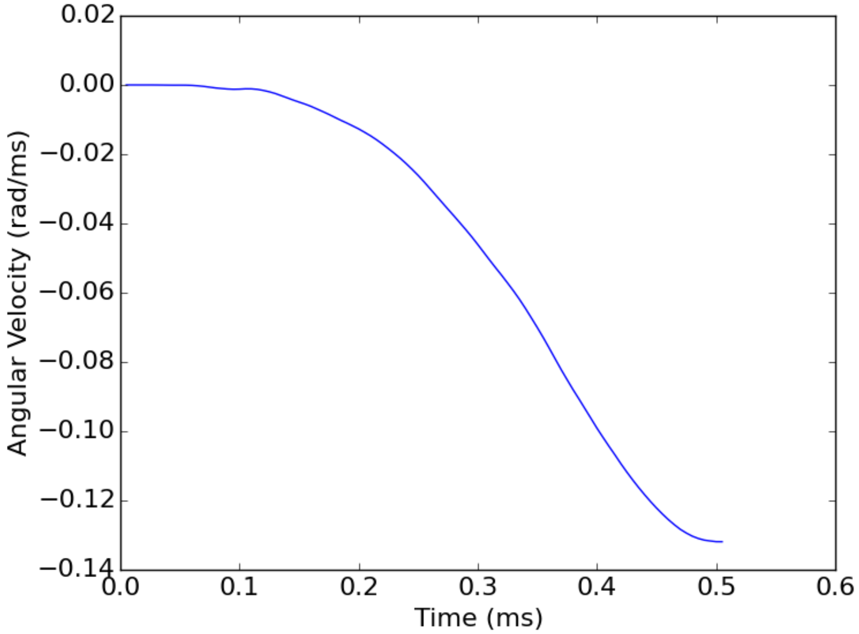


Figure 8.4.3: Angular velocity of the projectile about the x axis. The angular velocity is negative due to the orientation.

the weave. The increase in current in the yarns near the projectile is visible. The motion of the projectile pushes the yarns through the magnetic field in a way that causes a beneficial back electromotive force. This increases the current running through the yarns in contact with the projectile. The back EMF effect is not symmetrical. The yarns on the $-y$ side of the projectile are able slip in the direction they are forced in, decreasing the back EMF. The current running through three different yarns at different positions in the fabric is show in Figure 8.4.4 illustrating this effect. The asymmetric current distribution causes the $+y$ side of the fabric to be hotter than the $-y$, as seen in Figure 8.4.5.

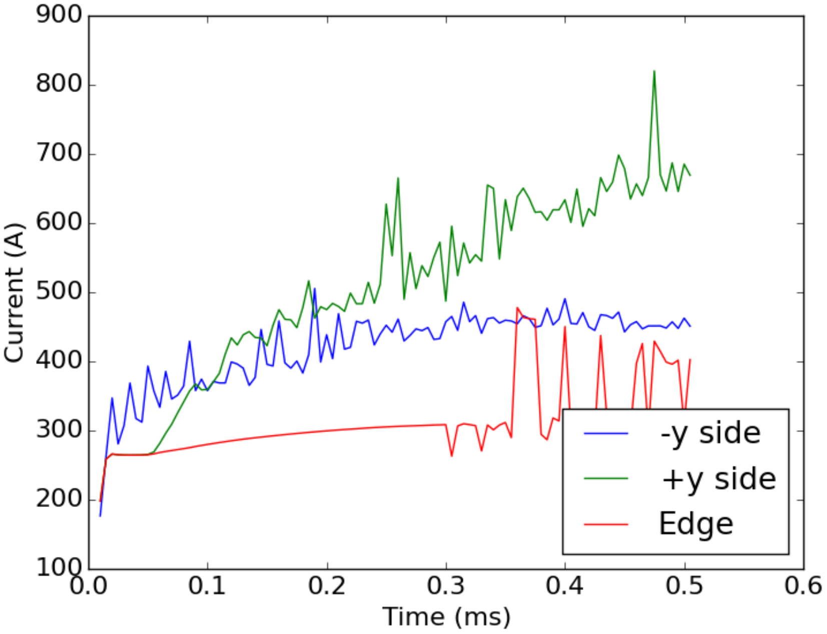


Figure 8.4.4: Electric current sampled in three different yarns, one on the $-y$ side of the projectile, one on the $+y$ side of the projectile, and one on the edge not in contact with the projectile.

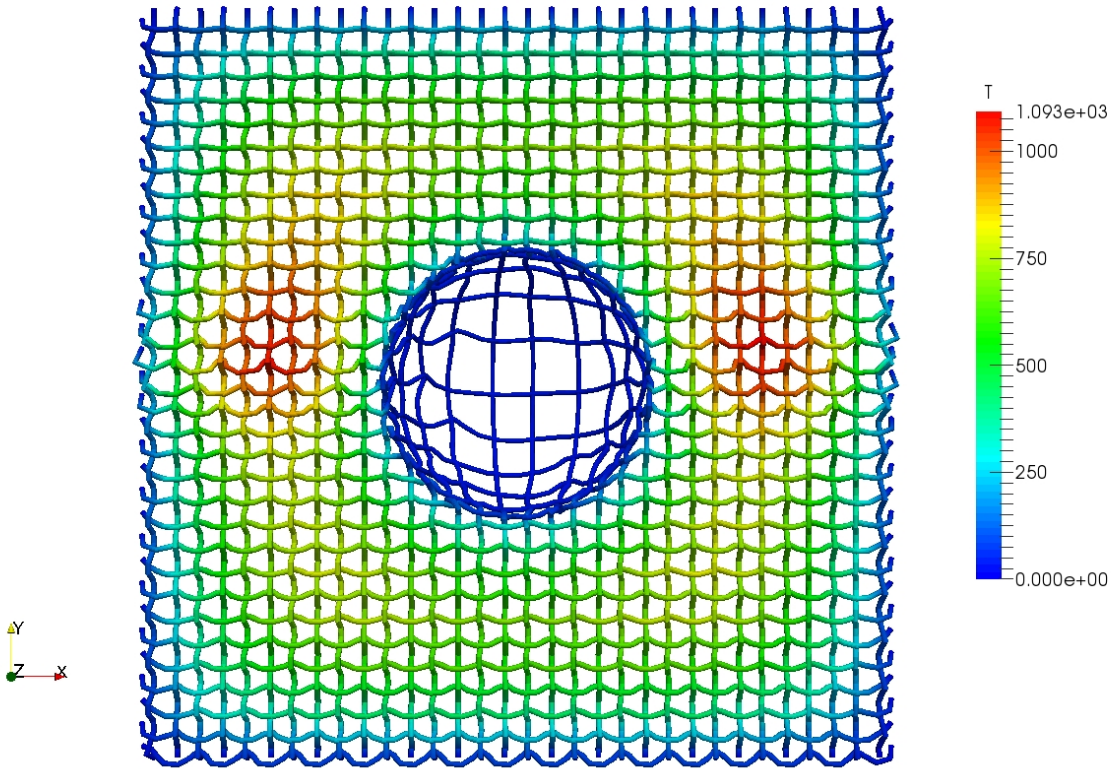


Figure 8.4.5: Temperature distribution at $t = 0.5ms$

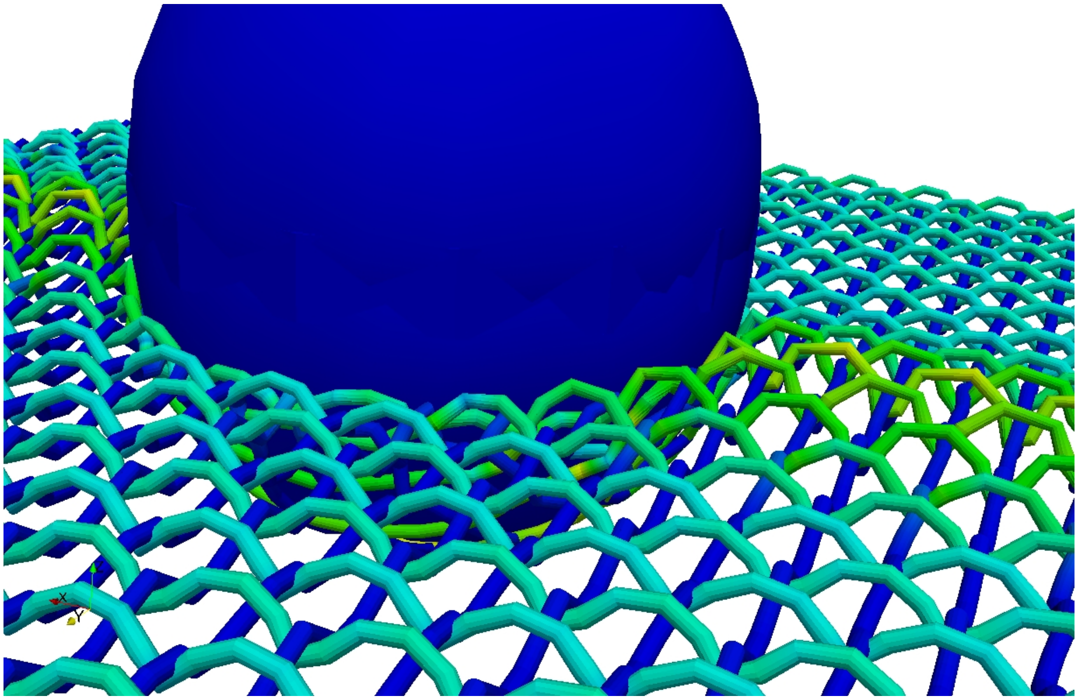


Figure 8.4.6: Close up near impact site at $t = 0.5ms$. The yarns are colored by current using the same scaling as Figure 8.4.1. The view is looking in the $-y$ direction.

Chapter 9

Outlook

9.1 Conclusion

A mixed finite element formulation for nonlinear beams incorporating thermal and electromagnetic fields was developed. The formulation was observed to solve static problems with the expected order of convergence using standard nodal basis functions of up to fourth order. The fully-coupled time dependent differential algebraic equation was solved using diagonally implicit Runge-Kutta methods with a monolithic Newton iteration to solve the stages. The convergence rates expected for standard ordinary differential equations for the three methods used were observed in all fields, including the implicit electrical potential, validating the solution scheme. The tangents of the multi-field mixed formulation required for the implicit scheme were generated automatically by employing a Domain Specific Language. Quadratic converge was observed in the applications of Newton's method using the code generated for the contact integrals. The framework was used to predict the mechanical, electrical, and thermal properties of a representative volume element of a plain woven fabric of conductive elements.

The usage of a high level domain specific language for differentiation code generation was crucial to this work. For some of the more complicated weak forms used, the generated C++ files could be over 10,000 lines long, with some individual lines exceeding 2,000 characters! Developing the local matrix tabulation routines for both the load vectors and tangent stiffnesses using only linear shape functions would have taken many more months of programming time. Linearizing the complex multiphysics contacts with higher order elements would have been intractable. Tweaks to the theoretical formulation were made constantly throughout the course of this work and could be tested immediately by only waited for the DSL specification to be recompiled. The expressions and their linearizations could take up to half an hour to compute and compile from scratch, but this is much faster development cycle than modifying and debugging low-level program by hand. The cost to this increased productivity is paid in learning curve to developing new features into an existing code. A few months were devoted to learning the structure of FFC and doflin.

9.2 Future Work

9.2.1 Parallelization

In addition to utilizing advanced mathematical code generation technologies, maximizing the usage of high performance computing resources requires parallelization. Developing using an existing, parallelized code—as opposed to writing a new scientific code from scratch—was a conscious design decision to take advantage of existing libraries. Most of the FEniCS package and the libraries it uses are parallelized, so that only the newly written code related to the contact handling needs to be altered. As with most prototype codes, the current implementation is not yet parallelized. Specifically, the following three procedures related to need to be updated:

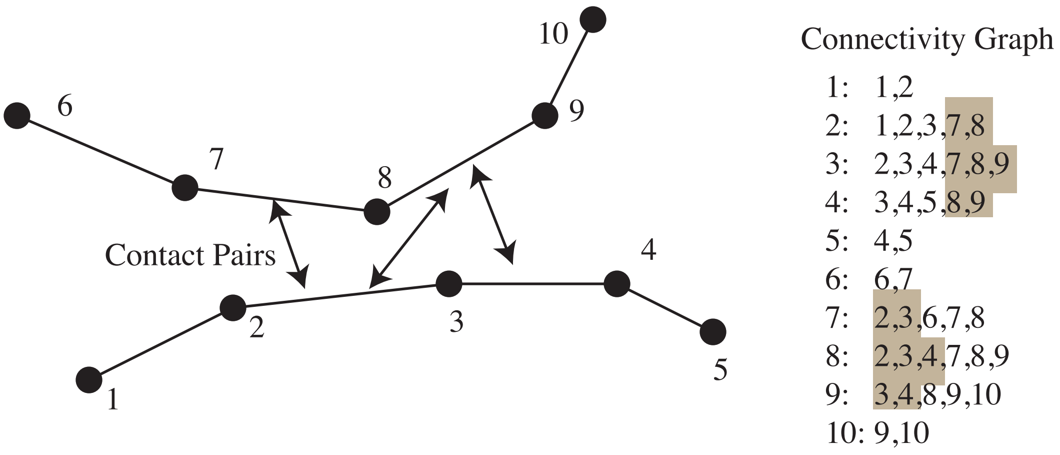
1. Searching for potentially contacting element pairs
2. Partitioning of the domain of contacting meshes
3. Assembling the global matrix and vector

The main challenge to parallelization of the fibril problem is partitioning the domain while preserving the ability to compute new contacts. Unlike a standard finite element problem, the connectivity for the fibril microstructure is made of the individual meshes *and* the contacts between them. The contact connectivity cannot be represented as a non-overlapping mesh. The contact pattern is able to change drastically during the course of a calculation and needs to be recalculated frequently.

A good parallelization scheme needs to both (1) minimize the total communication between connected processors while also (2) minimizing number of communicating processor pairs. Because of the long and thin nature of the fibrils, one fibril spans across most of the physical domain. This makes distributing intact fibrils across meshes a very bad choice, and it almost maximizes the communication required between processors. For a plain weave, one processor would be communicating with half of the processors! The contacting element pairs need to be considered in the connectivity graph, illustrated in Figure 9.2.1.

Because of the changing nature of the connectivity graph, a processor must store a larger portion of the graph than the standard overlapping degrees of freedom. The connectivity graph needs to contain element pairs that are not yet in contact but may be so in the future so that a processor contains the required portion of the mesh to perform the geometric search at the next point. That is, in addition to the overlapping nodes, data that is shared by adjacent processors, a "contact candidate" region must also be stored to perform the contact searches along the borders. The regions are illustrated in Figure 9.2.2. A bootstrapping issue exists where the processors need to perform the initial contact search before the first valid partition can be made. The initial guess partition needs to create a large overlap region to solve this.

The FEniCS package has mesh partitioning built-in using the libraries SCOTCH or ParMETIS. The matrix and vector wrappers provided by FEniCS were used for all of the calculations in this work. In the spirit of maximizing code reuse, no attempt was made to configure "what works" and the default backend uBLAS was used. Parallelized linear algebra backends are available once they are installed. The current prototype implementation of the contact assembly uses a partition in which the one processor owns the entire problem when initializing the matrix objects. The existing parallel matrix libraries can be used after the



Connectivity Graph

1: 1,2
2: 1,2,3,7,8
3: 2,3,4,7,8,9
4: 3,4,5,8,9
5: 4,5
6: 6,7
7: 2,3,6,7,8
8: 2,3,4,7,8,9
9: 3,4,8,9,10
10: 9,10

Figure 9.2.1: Connectivity graph of beam meshes and contacts. Entries due to contacts are highlighted.

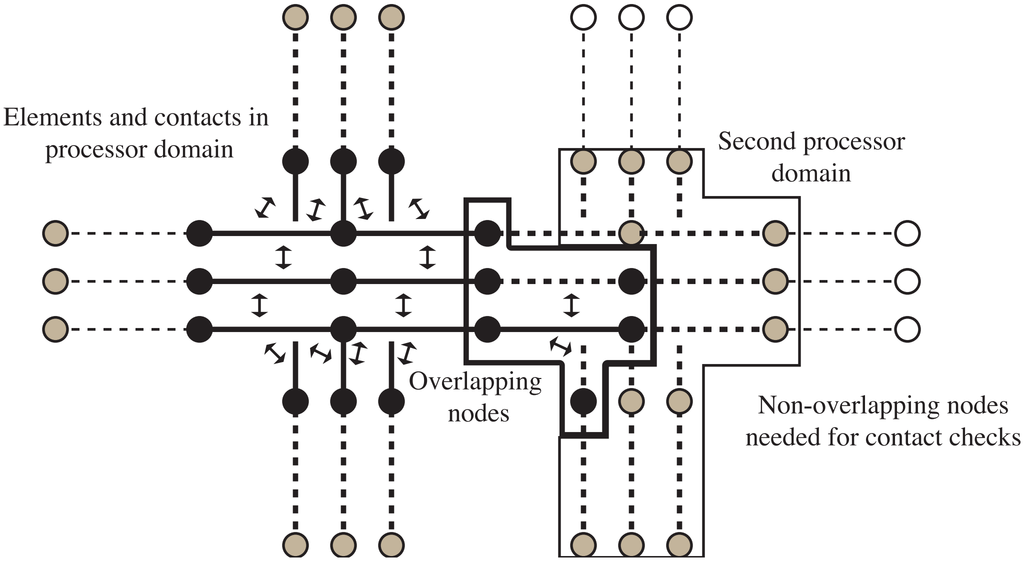

Elements and contacts in processor domain
Second processor domain
Overlapping nodes
Non-overlapping nodes needed for contact checks

Figure 9.2.2: Data management for partitioning scheme for contacts. The processor is responsible for calculations on elements drawn in solid lines and nodes it stores are filled. In addition to the nodes shared by elements in neighboring processors, a processor needs to store elements and nodes it needs to perform the geometric calculations to search for new contacts (shaded nodes.) Nodes that do not need to be stored are unfilled.

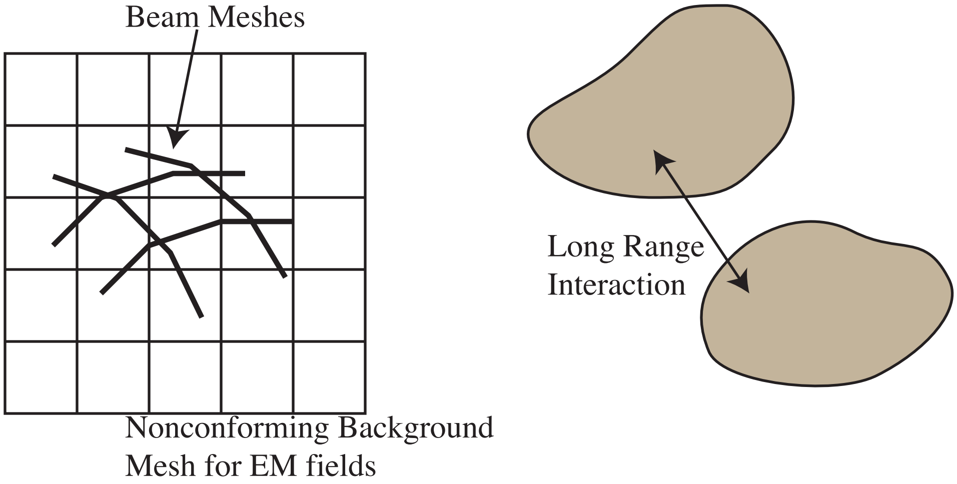


Figure 9.2.3: Potential non-contact applications of the added capacities to FEniCS. Left: coupling electromagnetic fields defined on a background mesh to the

contact search and partitioning is completed.

9.2.2 Physical Models

The two-element mapping integral framework that was developed can be applied to other problems besides contacts with some generalizations. The contact implementation is only able to handle coupling two of the same element. For example, there is no way to compute the contact between beams and the surface of a solid mesh. In addition to the geometric routines, this requires adding more syntax to the Unified Form Language and the FEniCS Form Compiler to express this type of two-element integral. The same contact framework can be used in other instances where coupling between finite element meshes is desired. For example, the formulation of beam problem can be extended to solve for the magnetic vector potential on a separate background mesh. The variational equation can be expressed as integrals over two different domains, with the domain of the material Ω_M a subset of that of space $\Omega_M \subset \Omega_S$,

$$\int_{\Omega_S} \nabla \times \delta\mathbf{A} \cdot \mu^{-1} \nabla \times \mathbf{A} d\Omega = \underbrace{\int_{\Omega_M} \delta\mathbf{A} \cdot \mathbf{j} d\Omega}_{\text{Both shape functions}}$$

with the fields constructed from different shape functions defined on the two meshes, e.g. $\mathbf{A} = \sum_A \hat{\mathbf{A}}_A \phi_A^S(x)$, $x \in \Omega_S$ and $\mathbf{j} = \sum_A -\hat{V}_A \sigma \nabla \phi_A^M(x)$ $x \in \Omega_M$. The framework can also be used to incorporate long-range interactions, such as molecular potentials, electromagnetic forces, and gravitational attractions. A pair-wise force interaction from electrical charges $\mathbf{f}(x) = \frac{q_1 q_2}{r^2}\mathbf{n}$ can be expressed as an integral across the two bodies by $\int\int [[\delta\mathbf{x}]] \cdot \frac{q^+ q^-}{[[\mathbf{x}]] \cdot [[\mathbf{x}]]} [[\mathbf{x}]] \, d\Omega^+ d\Omega^-$.

The framework developed is general enough to incorporate a wide variety of constitutive responses. A number of improvements can be made to the constitutive laws for the applications shown. Failure and plasticity models are needed to perform in-depth analyses of ballistic armor. Pressure dependence conductivities can be used to improve the analysis of thermal and electrical resistances in the textile. Static friction has a significant effect on the energy dampening in the dynamic analysis as well as the effective shear strength of a textile in static analysis [17, 71, 70]. Many of the standard frictional analysis methods employ history data that lives on the contact mappings, however, which requires more code development in FEniCS to represent such a function. Non-constant contact laws can be implemented, such as short-range van-der Wals attractions.

Piezoelectric and other electromagnetically coupled constitutive laws can be used to analyze textile-based electromechanical devices. The thermomechanical coupling of the model can also be used to study thermally actuated textiles, such as the actuator illustrated in [52]. Heterogenous textiles employing fabrics with different coefficients of thermal expansion can be used to design “smart” clothing that reshapes in response to the wearer’s and environment’s temperature to adjust its insulating properties. Many biological tissues have fibrous microstructures and respond to electromagnetic fields, such as muscle tissue, for which this model is also applicable.

Bibliography

[1] Roger Alexander. Diagonally implicit Runge-Kutta methods for stiff ODE's. *SIAM Journal on Numerical Analysis*, 14(6):1006–1021, 1977.

[2] Francisco Armero. *Finite Element Methods: 1970's and Beyond*, chapter Assumed Strain Finite Element Methods. CIMNE, Barcelona, 2004.

[3] Hugues Bajas, Damien Durville, Daniel Ciazynski, and Arnaud Devred. Numerical simulation of the mechanical behavior of ITER cable-in-conduit conductors. *Applied Superconductivity, IEEE Transactions on*, 20(3):1467–1470, 2010.

[4] Matthew I Barham, Daniel A White, and David J Steigmann. Finite element modeling of the deformation of magnetoelastic film. *Journal of Computational Physics*, 229(18):6193–6207, 2010.

[5] EJ Barshaw, J White, MJ Chait, et al. High energy density (HED) biaxially-oriented poly-propylene (BOPP) capacitors for pulse power applications. *Magnetics, IEEE Transactions on*, 43(1):223–225, 2007.

[6] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, et al. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*, 2012.

[7] L.A.A. Beex. *Multiscale quasicontinuum modelling of fibrous materials.* PhD thesis, Technische Universiteit Eindhoven, 2012.

[8] Ted Belytschko, Wing Kam Liu, Brian Moran, and Khalil Elkhodary. *Nonlinear finite elements for continua and structures.* John Wiley & Sons, 2013.

[9] James Bergstra, Olivier Breuleux, Frédéric Bastien, et al. Theano: A CPU and GPU math compiler in Python. In *Proc. 9th Python in Science Conf*, pages 1–7, 2010.

[10] Joanna Berzowska and Marguerite Bromley. Soft computation through conductive textiles. In *Proceedings of the International Foundation of Fashion Technology Institutes Conference*, pages 12–15, 2007.

[11] M Besbes, Z Ren, and A Razek. Finite element analysis of magneto-mechanical coupled phenomena in magnetostrictive materials. *Magnetics, IEEE Transactions on*, 32(3):1058–1061, 1996.

[12] HH Billon and DJ Robinson. Models for the ballistic impact of fabric armour. *International Journal of Impact Engineering*, 25(4):411–422, 2001.

[13] P. Boisse, A. Gasser, B. Hagege, and J.L. Billoet. Analysis of the mechanical behavior of woven fibrous material using virtual tests at the unit cell level. *Journal of materials science*, 40(22):5955–5962, 2005.

[14] J. Bonet and R.D. Wood. *Nonlinear continuum mechanics for finite element analysis.* Cambridge Univ Pr, 1997.

[15] B.B. Boubaker, B. Haussy, and J.F. Ganghoffer. Consideration of the yarn-yarn interactions in meso/macro discrete model of fabric. Part I: Single yarn behaviour. *Mechanics Research Communications*, 34(4):359–370, 2007.

[16] Gerd Brandstetter and Sanjay Govindjee. A high-order immersed boundary discontinuous-Galerkin method for Poisson's equation with discontinuous coefficients and singular sources. *International Journal for Numerical Methods in Engineering*, 2014.

[17] BJ Briscoe and F. Motamedi. The ballistic impact characteristics of aramid fabrics: the influence of interface friction. *Wear*, 158(1):229–247, 1992.

[18] Leah Buechley and Michael Eisenberg. The lilypad arduino: Toward wearable engineering for everyone. *Pervasive Computing, IEEE*, 7(2):12–15, 2008.

[19] Leah Buechley and Michael Eisenberg. Fabric PCBs, electronic sequins, and socket buttons: techniques for e-textile craft. *Personal and Ubiquitous Computing*, 13(2):133–150, 2009.

[20] Leah Buechley, Sue Hendrix, and Mike Eisenberg. Paints, paper, and programs: first steps toward the computational sketchbook. In *Proceedings of the 3rd International Conference on Tangible and Embedded Interaction*, pages 9–12. ACM, 2009.

[21] Leah Buechley and Benjamin Mako Hill. LilyPad in the wild: how hardware's long tail is supporting new engineering and design communities. In *Proceedings of the 8th ACM Conference on Designing Interactive Systems*, pages 199–207. ACM, 2010.

[22] Mario Cannarozzi. A minimum principle for tractions in the elastostatics of cable networks. *International journal of solids and structures*, 23(5):551–568, 1987.

[23] Yanjie Cao, Chengxue Wang, Huijin Wang, and Hongbo Jin. Simulation of electromagnetic launcher of active electromagnetic armor. In *Electromagnetic Launch Technology, 2008 14th Symposium on*, pages 1–4. IEEE, 2008.

[24] Yuen Aoi Chee, Azam Ahmad Bakir, and Dedy HB Wicaksono. Proprioceptive sensing system for therapy assessment using textile-based biomedical Micro Electro Mechanical System (MEMS). In *Sensors, 2012 IEEE*, pages 1–4. IEEE, 2012.

[25] B.A. Cheeseman and T.A. Bogetti. Ballistic impact into fabric and compliant composite laminates. *Composite Structures*, 61(1):161–173, 2003.

[26] Kunigunde Cherenack, Christoph Zysset, Thomas Kinkeldei, Niko Münzenrieder, and Gerhard Tröster. Woven electronic fibers with sensing and display functions for smart textiles. *Advanced Materials*, 22(45):5178–5182, 2010.

[27] I.S. Chocron-Benloulo, J. Rodriguez, and V. Sanchez-Galvez. A simple analytical model to simulate textile fabric ballistic impact behavior. *Textile research journal*, 67(7):520–528, 1997.

[28] IS Chocron Benloulo and V. Sanchez-Galvez. A new analytical model to simulate impact onto ceramic/composite armors. *International journal of impact engineering*, 21(6):461–471, 1998.

[29] Peter W Chung, Kumar K Tamma, and Raju R Namburu. A micro/macro homogenization approach for viscoelastic creep analysis with dissipative correctors for heterogeneous woven-fabric layered media. *Composites science and technology*, 60(12):2233–2253, 2000.

[30] Cédric Cochrane, Vladan Koncar, Maryline Lewandowski, and Claude Dufour. Design and development of a flexible strain sensor for textile structures based on a conductive polymer composite. *Sensors*, 7(4):473–492, 2007.

[31] George Albert Costello. *Theory of wire rope*. Springer Verlag, 1997.

[32] JP Coyette and P Guisset. Cable network analysis by a nonlinear programming technique. *Engineering Structures*, 10(1):41–46, 1988.

[33] NV David, X-L Gao, and JQ Zheng. Ballistic resistant body armor: contemporary and prospective materials and related protection mechanisms. *Applied Mechanics Reviews*, 62:050802, 2009.

[34] Zachary DeVito and Pat Hanrahan. Designing the language liszt for building portable mesh-based pde solvers. In *SciDAC 2011 Conference*. Citeseer, 2011.

[35] Zachary DeVito, James Hegarty, Alex Aiken, Pat Hanrahan, and Jan Vitek. Terra: a multi-stage language for high-performance computing. In *ACM SIGPLAN Notices*, volume 48, pages 105–116. ACM, 2013.

[36] Zachary DeVito, Niels Joubert, Francisco Palacios, et al. Liszt: a domain specific language for building portable mesh-based PDE solvers. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, page 9. ACM, 2011.

[37] Y. Duan, M. Keefe, TA Bogetti, BA Cheeseman, and B. Powers. A numerical investigation of the influence of friction on energy absorption by a high-strength fabric subjected to ballistic impact. *International journal of impact engineering*, 32(8):1299–1312, 2006.

[38] D. Durville. Contact-friction modeling within elastic beam assemblies: an application to knot tightening. *Computational Mechanics*, pages 1–21, 2012.

[39] Damien Durville. Modelling of contact-friction interactions in entangled fibrous materials. In *VI World Wide Congress on Computational Mechanics*, 2004.

[40] Damien Durville. Finite element simulation of textile materials at mesoscopic scale. In *Finite element modelling of textiles and textile composites*, 2007.

[41] Damien Durville. A finite element approach of the behaviour of woven materials at microscopic scale. In *Mechanics of microstructured solids*, pages 39–46. Springer, 2009.

[42] George F Eichinger, Kara Baumann, Thomas Martin, and Mark Jones. Using a PCB layout tool to create embroidered circuits. In *Wearable Computers, 2007 11th IEEE International Symposium on*, pages 105–106. IEEE, 2007.

[43] Sebastian Fillep, Julia Mergheim, and Paul Steinmann. Computational homogenization of rope-like technical textiles. *Computational Mechanics*, 55(3):577–590, 2015.

[44] Limor Fried. FLORA - Wearable electronic plateform. http://www.adafruit.com/product/659, November 2012.

[45] James Gere. *Mechanics of Materials*. Cenage Learning, Toronto, 6th edition, 2006.

[46] TK Ghosh, A Dhawan, and JF Muth. Formation of electrical circuits in textile structures. *Intelligent textiles and clothing*, pages 239–282, 2006.

[47] Scott Gilliland, Nicholas Komor, Thad Starner, and Clint Zeagler. The Textile Interface Swatchbook: Creating graphical user interface-like widgets with conductive embroidery. In *Wearable Computers (ISWC), 2010 International Symposium on*, pages 1–8. IEEE, 2010.

[48] Michael Griebel, Thomas Dornseifer, and Tilman Neunhoeffer. *Numerical simulation in fluid dynamics: a practical introduction*, volume 3. Siam, 1997.

[49] Andreas Griewank et al. On automatic differentiation. *Mathematical Programming: recent developments and applications*, 6(6):83–107, 1989.

[50] B. Gu. Ballistic penetration of conically cylindrical steel projectile into plain-woven fabric target–a finite element simulation. *Journal of composite materials*, 38(22):2049–2074, 2004.

[51] Heinrich Guggenheimer. *Differential Geometry*. McGraw-Hill, New York, 1963.

[52] Carter S Haines, Márcio D Lima, Na Li, et al. Artificial muscles from fishing line and sewing thread. *science*, 343(6173):868–872, 2014.

[53] Mahiar Hamedi, Robert Forchheimer, and Olle Inganäs. Towards woven logic from organic electronic fibres. *Nature materials*, 6(5):357–362, 2007.

[54] Mir Mohammad Badrul Hasan, Olaf Diestel, and Chokri Cherif. Electro-mechanical properties of friction spun conductive hybrid yarns made of carbon filaments for composites. *Textile Research Journal*, 81(15):1603–1616, 2011.

[55] EM Haseganu and DJ Steigmann. Analysis of partly wrinkled membranes by the method of dynamic relaxation. *Computational Mechanics*, 14(6):596–614, 1994.

[56] Amy Henderson, Jim Ahrens, Charles Law, et al. *The ParaView Guide*. Kitware Clifton Park, NY, 2004.

[57] Pavel Holoborodko. Cubature formulas for the unit disk. http://www.holoborodko.com/pavel/numerical-methods/numerical-integration/cubature-formulas-for-the-unit-disk/, August 2010.

[58] Thomas JR Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.

[59] Charles R Hummer. Inductance of parallel plates in electromagnetic armor. Technical report, DTIC Document, 2006.

[60] John D Hunter. Matplotlib: A 2D graphics environment. *Computing in science and engineering*, 9(3):90–95, 2007.

[61] Umran S Inan and Robert A Marshall. *Numerical electromagnetics: the FDTD method*. Cambridge University Press, 2011.

[62] Frank P Incropera. *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.

[63] John David Jackson. Classical electrodynamics. *Classical Electrodynamics, 3rd Edition, by John David Jackson, pp. 832. ISBN 0-471-30932-X. Wiley-VCH, July 1998.*, 1, 1998.

[64] Hrvoje Jasak, Aleksandar Jemcov, and Zeljko Tukovic. OpenFOAM: A C++ library for complex physics simulations. 2013.

[65] K. L. Johnson. *Contact Mechanics*. Cambridge University Press, August 1987.

[66] Eric Jones, Travis Oliphant, and Pearu Peterson. SciPy: Open source scientific tools for Python. 2014.

[67] J.M. Jovicic. *Numerical modeling and analysis of static and ballistic behavior of multilayered/multiphase composite materials using detailed microstructural discretization*. PhD thesis, Drexel University, 2003.

[68] B Karaguzel, CR Merritt, T Kang, et al. Flexible, durable printed electrical circuits. *The Journal of the Textile Institute*, 100(1):1–9, 2009.

[69] Rakesh B Katragadda and Yong Xu. A novel intelligent textile technology based on silicon flexible skins. *Sensors and Actuators A: Physical*, 143(1):169–174, 2008.

[70] John E Kirkwood, Keith M Kirkwood, Young Sil Lee, et al. Yarn Pull-Out as a Mechanism for Dissipating Ballistic Impact Energy in Kevlar® KM-2 Fabric Part II: Predicting Ballistic Performance. *Textile research journal*, 74(11):939–948, 2004.

[71] K.M. Kirkwood, J.E. Kirkwood, Y.S. Lee, et al. Yarn Pull-Out as a Mechanism for Dissipating Ballistic Impact Energy in Kevlar® KM-2 Fabric Part I: Quasi-Static Characterization of Yarn Pull-Out. *Textile research journal*, 74(10):920–928, 2004.

[72] Attay Kovetz. *Electromagnetic theory.* Oxford University Press Oxford, 2000.

[73] A Krajewski, Kevin Magniez, R Helmer, and Viktoria Schrank. Piezoelectric force response of novel 2D textile based PVDF sensors. *Sensors Journal, IEEE*, 13(12):4743–4748, Dec 2013.

[74] Natasha C Lagoudas and Zoubeida Ounaies. Electrospinning of continuous piezoelectric yarns for composite application. In *The 15th International Symposium on: Smart Structures and Materials & Nondestructive Evaluation and Health Monitoring*, pages 69292N–69292N. International Society for Optics and Photonics, 2008.

[75] Cédric P Laurent, Damien Durville, Didier Mainard, Jean-François Ganghoffer, and Rachid Rahouadj. A multilayer braided scaffold for Anterior Cruciate Ligament: Mechanical modeling at the fiber scale. *Journal of the mechanical behavior of biomedical materials*, 12:184–196, 2012.

[76] Josephine B Lee and Vivek Subramanian. Organic transistors on fiber: a first step towards electronic textiles. In *Electron Devices Meeting, 2003. IEDM'03 Technical Digest. IEEE International*, pages 8–3. IEEE, 2003.

[77] Josephine B Lee and Vivek Subramanian. Weave patterned organic transistors on fiber for e-textiles. *Electron Devices, IEEE Transactions on*, 52(2):269–275, 2005.

[78] Y.S. Lee, E.D. Wetzel, and N.J. Wagner. The ballistic impact characteristics of Kevlar® woven fabrics impregnated with a colloidal shear thickening fluid. *Journal of materials science*, 38(13):2825–2833, 2003.

[79] CM Leech and BA Adeyefa. Dynamics of cloth subject to ballistic impact velocities. *Computers & Structures*, 15(4):423–432, 1982.

[80] Randall J LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*, volume 98. Siam, 2007.

[81] ZH Lim, ZX Chia, M Kevin, ASW Wong, and GW Ho. A facile approach towards ZnO nanorods conductive textile for room temperature multifunctional sensors. *Sensors and Actuators B: Chemical*, 151(1):121–126, 2010.

[82] Torsten Linz, Lena Gourmelon, and Geert Langereis. Contactless EMG sensors embroidered onto textile. In *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*, pages 29–34. Springer, 2007.

[83] Torsten Linz, Christine Kallmayer, Rolf Aschenbrenner, and Herbert Reichl. Embroidering electrical interconnects with conductive yarn for the integration of flexible electronic modules into fabric. In *Wearable Computers, 2005. Proceedings. Ninth IEEE International Symposium on*, pages 86–89. IEEE, 2005.

[84] Torsten Linz, Erik Simon, and Hans Walter. Fundamental analysis of embroidered contacts for electronics in textiles. In *Electronic System-Integration Technology Conference (ESTC), 2010 3rd*, pages 1–5. IEEE, 2010.

[85] Anders Logg, Kent-Andre Mardal, and Garth Wells. *Automated solution of differential equations by the finite element method: The FEniCS book*, volume 84. Springer Science & Business Media, 2012.

[86] Anders Logg, Kristian B Ølgaard, Marie E Rognes, and Garth N Wells. Ffc: the fenics form compiler. In *Automated Solution of Differential Equations by the Finite Element Method*, pages 227–238. Springer, 2012.

[87] Anders Logg, Garth N Wells, and Johan Hake. DOLFIN: A C++/Python finite element library. In *Automated Solution of Differential Equations by the Finite Element Method*, pages 173–225. Springer, 2012.

[88] Tomasz Maleszka and Pawel Kabacik. Bandwidth properties of embroidered loop antenna for wearable applications. In *Wireless Technology Conference (EuWIT), 2010 European*, pages 89–92. IEEE, 2010.

[89] Dina Meoli and Traci May-Plumlee. Interactive electronic textile development: A review of technologies. *Journal of Textile and Apparel, Technology and Management*, 2(2):1–12, 2002.

[90] Yuyang Miao, Eric Zhou, Youqi Wang, and Bryan A Cheeseman. Mechanics of textile composites: Micro-geometry. *Composites Science and Technology*, 68(7):1671–1678, 2008.

[91] B. Nadler, P. Papadopoulos, and D.J. Steigmann. Multiscale constitutive modeling and numerical simulation of fabric material. *International journal of solids and structures*, 43(2):206–221, 2006.

[92] Zahi Nakad, Mark T Jones, and Thomas Martin. Communications in Electronic Textile Systems. In *Communications in Computing*, pages 37–46, 2003.

[93] CA Navarro. Simplified Modelling of the Ballistic Behaviour of Fabrics and Fibre-Reinforced Polymetric Matrix Composites. *Key Engineering Materials*, 141:383–402, 1997.

[94] Jean-Claude Nédélec. Mixed finite elements in 3. *Numerische Mathematik*, 35(3):315–341, 1980.

[95] G. Nilakantan, M. Keefe, T.A. Bogetti, R. Adkinson, and J.W. Gillespie. On the finite element analysis of woven fabric impact using multiscale modeling techniques. *International Journal of Solids and Structures*, 47(17):2300–2315, 2010.

[96] Gaurav Nilakantan and Steven Nutt. Effects of clamping design on the ballistic impact response of soft body armor. *Composite Structures*, 108:137–150, 2014.

[97] AR Othman and MH Hassan. Effect of Different Construction Designs of Aramid Fabric on the Ballistic Performances. *Materials & Design*, 2012.

[98] B. Parga-Landa and F. Hernandez-Olivares. An analytical model to predict impact behaviour of soft armours. *International journal of impact engineering*, 16(3):455–466, 1995.

[99] Pankaj K Porwal and S Leigh Phoenix. Modeling system effects in ballistic impact into multi-layered fibrous materials for soft body armor. *International journal of fracture*, 135(1):217–249, 2005.

[100] E Rehmi Post, Maggie Orth, PR Russo, and Neil Gershenfeld. E-broidery: Design and fabrication of textile-based computing. *IBM Systems Journal*, 39(3.4):840–860, 2000.

[101] Alejandro Queiruga. A Computational Model of Electromagnetically Sensitive Fabric Armor Colliding with a Rigid Projectile. Master's thesis, University of California, Berkeley, 2013.

[102] Z Ren, B Ionescu, M Besbes, and A Razek. Calculation of mechanical deformation of magnetic materials in electromagnetic devices. *Magnetics, IEEE Transactions on*, 31(3):1873–1876, 1995.

[103] MB Rubin. *Cosserat theories: Shells, rods and points*, volume 79 of *Solid Mechanics and its Applications*. Kluwer Academic Publishers, Dordrecht, 2000.

[104] MB Rubin. On the theory of a Cosserat point and shear locking in thin beams. *Communications in numerical methods in engineering*, 17(3):201–213, 2001.

[105] VPW Shim, YB Guo, and VBC Tan. Response of woven and laminated high-strength fabric to oblique impact. *International Journal of Impact Engineering*, 48:87–97, 2012.

[106] D.A. Shockey, D.C. Erlich, and J.W. Simons. Lightweight fragment barriers for commercial aircraft. In *Proceedings of the 18th International Symposium on Ballistics, San Antonio, Texas*, 1999.

[107] David J Steigmann. On the formulation of balance laws for electromagnetic continua. *Mathematics and Mechanics of Solids*, 2007.

[108] Klaus Sterzelmeier, G Buderer, C Ganthier-Blum, and W Wenning. Electromagnetic armor test facility with modular pulsed power conceptual design. In *Pulsed Power Conference, 1999. Digest of Technical Papers. 12th IEEE International*, volume 2, pages 1361–1364. IEEE, 1999.

[109] ALA Tabiei and G. Nilakantan. Ballistic impact of dry woven fabric composites: a review. *Applied Mechanics Reviews*, 61:010801, 2008.

[110] VBC Tan and TW Ching. Computational simulation of fabric armour subjected to ballistic impacts. *International journal of impact engineering*, 32(11):1737–1751, 2006.

[111] Xiaoming Tao. Electrical textile sensors for repeated large deformation: structures and electromechanical properties. In *International Conference on Experimental Mechanics 2008 and Seventh Asian Conference on Experimental Mechanics*, pages 73754H–73754H. International Society for Optics and Photonics, 2008.

[112] Alexandre Torre, Hugues Bajas, D Ciazynski, Damien Durville, and K Weiss. Mechanical-Electrical Modeling of Stretching Experiment on 45 Strands CICCs. *Applied Superconductivity, IEEE Transactions on*, 21(3):2042–2045, 2011.

[113] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The NumPy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[114] Derman Vatansever, Elias Siores, Ravi L Hadimani, and Tahir Shah. Smart Woven Fabrics in Renewable Energy Generation.

[115] JR Vinson and JA Zukas. On the ballistic impact of textile body armor. *Journal of Applied mechanics*, 42:263, 1975.

[116] Kai Wang, Qinghai Meng, Yajie Zhang, Zhixiang Wei, and Menghe Miao. High-performance two-ply yarn supercapacitors based on carbon nanotubes and polyaniline nanowire arrays. *Advanced materials*, 25(10):1494–1498, 2013.

[117] Youqi Wang, Yuyang Miao, Daniel Swenson, et al. Digital element approach for simulating impact and penetration of textiles. *International Journal of Impact Engineering*, 37(5):552–560, 2010.

[118] Youqi Wang and Xuekun Sun. Digital-element simulation of textile processes. *Composites science and technology*, 61(2):311–319, 2001.

[119] Matthias Wickert. Electric armor against shaped charges: Analysis of jet distortion with respect to jet dynamics and current flow. *Magnetics, IEEE Transactions on*, 43(1):426–429, 2007.

[120] Carole A Winterhalter, Justyna Teverovsky, Patricia Wilson, et al. Development of electronic textiles to support networks, communications, and medical applications in future US Military protective clothing systems. *IEEE Transactions on Information Technology in Biomedicine*, 9(3):402–406, 2005.

[121] Inc. Wolfram Research. *Mathematica*. Wolfram Research, Inc., Champaign, Illinois, version 9.0 edition, 2012.

[122] Li Xiaopeng, Meng Tao, Zhao Chun, and Li Liyi. Multiprojectile active electromagnetic armor. *Magnetics, IEEE Transactions on*, 43(1):460–462, 2007.

[123] P Xue, XM Tao, Keith WY Kwok, MY Leung, and TX Yu. Electromechanical behavior of fibers coated with an electrically conductive polymer. *Textile research journal*, 74(10):929–936, 2004.

[124] Hui Zhang, Xiaoming Tao, Shanyuan Wang, and Tongxi Yu. Electro-mechanical properties of knitted fabric made from conductive multi-filament yarn under unidirectional extension. *Textile research journal*, 75(8):598–606, 2005.

[125] Hui Zhang, Xiaoming Tao, Tongxi Yu, and Shanyuan Wang. Conductive knitted fabric as large-strain gauge under high temperature. *Sensors and Actuators A: Physical*, 126(1):129–140, 2006.

[126] Ping Zheng, Yong Liu, Shukang Cheng, Zhiyuan Li, and Jinsuo Hu. Research on the passive electromagnetic armor. *Magnetics, IEEE Transactions on*, 41(1):456–459, 2005.

[127] Li Zhiyuan, Ming Sun, Feng Ming, and Chao Yangjie. The finite element analysis for the magnetic of the active EM armor projectile interceptor. In *Electromagnetic Launch Technology, 2004. 2004 12th Symposium on*, pages 441–443. IEEE, 2005.

[128] E Zhou, DH Mollenhauer, and EV Larve. A realistic 3-D textile geometric model. In *Seventeenth international conference on composite materials ICCM-17, Edinburgh, United Kingdom*, 2009.

[129] Guangming Zhou, Xuekun Sun, and Youqi Wang. Multi-chain digital element analysis in textile mechanics. *Composites science and Technology*, 64(2):239–244, 2004.

[130] TI Zohdi. High-speed impact of electromagnetically sensitive fabric and induced projectile spin. *Computational Mechanics*, 46(3):399–415, 2010.

[131] TI Zohdi. Electromagnetically-induced deformation of functionalized fabric. *Journal of Elasticity*, 105(1):381–398, 2011.

[132] TI Zohdi and D. Powell. Multiscale construction and large-scale simulation of structural fabric undergoing ballistic impact. *Computer methods in applied mechanics and engineering*, 195(1-3):94–109, 2006.

Appendix A

Example Python Script

A.1 Setup

This file solves for the deformation of a textile in a B field.
It must be run from the root directory,

```
python programs/em_deformation.py
```

You will have to make sure the output directory exists,

```
mkdir post/em_deformation
```

Firstly, you need to import the library:

```
from src import *
from IPython import embed
```

Now, set up the geometry object:

```
sheets = [
    Geometries.PlainWeaveFibrils(8,12.5,10.0, 8,12.5,10.0,
                                 0.0,0.81, [ 1 ],0.81)
    ]

endpts = []
for s in sheets:
    endpts.extend( s.endpts() )
```

And define all fo the properties:

```
E = 1.26 #MPa
nu = 0.0
phi = 0.0 #np.pi/4.0
defaults = { 'mu':E/(2*(1 + nu)),
             'lambda': E*nu/((1 + nu)*(1 - 2*nu)),
             'rho':0.00144,
             'radius':0.4,
             'em_B':Constant( (0.005*np.sin(phi),0.005*np.cos(phi),0.0)),
             'contact_penalty': 500.0,
             'dissipation':0.01,
             'contact_em': 1.0,
             'em_bc_r_0': 100.0,
             'em_bc_J_0': -1000.0
             }
props = [ {} for i in endpts ]
for i in xrange(sheets[0].NX,sheets[0].NX+sheets[0].NY):
    props[i]['radius'] = 0.6
    props[i]['em_bc_r_0'] = 0.001
    props[i]['em_bc_J_0'] = 0.0
Nelems = [ 30 for i in endpts ]
```

A.2 Data structure allocations

Initialzie the warp object on the MonolothicProblem object

```

warp = Warp(endpts,props,defaults, Nelems, MonolithicProblem, order = (1,1))
```

Utility function for output:

```
outputcnt = 0
def output():
    global outputcnt
    warp.output_states("post/em_deformation/yarn_{0}_"+str(outputcnt)+".pvd",1)
    warp.output_solids("post/em_deformation/mesh_{0}_"+str(outputcnt)+".pvd",1)
    outputcnt+=1
output()
```

Loop through all of the geometry objects and apply their initial geometries. Initialize the Voltage and Temperature fields as well.

```
istart=0
for s in sheets:
    s.initialize(warp,istart)
    istart += s.nfibril
warp.pull_fibril_fields()
for fib in warp.fibrils:
    temp_field = Function(fib.problem.spaces['S'])
    temp_field.interpolate(Expression("A*x[0]+B",A=0.0/sheets[0].restX,B=0.0))
    assign(fib.problem.fields['wv'].sub(4), temp_field)
    temp_field.interpolate(Expression("1.0"))
    assign(fib.problem.fields['wv'].sub(3), temp_field)
warp.pull_fibril_fields()
warp.update()
```

Create the contact pairs for all of the geometry objects and initialize the contact group data structure.

```
cpairs = []
for s in sheets:
    cpairs.extend(s.contact_pairs())
cpairs.extend((i,len(endpts)-1) for i in xrange(len(endpts)-1))
warp.create_contacts(pairs=cpairs,cutoff=1.5)

output()
```

A.3 Time stepper setup

Set up all of the boundary conditions.

```
bound_all = CompiledSubDomain("on_boundary")
bound_sides = CompiledSubDomain(
    "( near(x[0],y) || near(x[0],-y) ) && on_boundary",
    y=sheets[0].restX)
bound_one_side = CompiledSubDomain("( near(x[0],y) ) && on_boundary",
                                   y=sheets[0].restX)

subq = MultiMeshSubSpace(warp.spaces['W'],0)
subT = MultiMeshSubSpace(warp.spaces['W'],3)
subVol = MultiMeshSubSpace(warp.spaces['W'],4)
zeroV = Constant((0.0,0.0,0.0))
zeroS = Constant(0.0)
bcq = MultiMeshDirichletBC(subq,zeroV, bound_all)
bcT = MultiMeshDirichletBC(subT,zeroS, bound_all)
bcVol = MultiMeshDirichletBC(subVol,zeroS, bound_one_side)
```

Set up the time stepper object and the functions to pass it.

```
def apply_BCs(K,R,t,hold=False):
    bcq.apply(K,R)
    bcT.apply(K,R)
    bcVol.apply(K,R)
def sys(time):
    return warp.assemble_forms(['F','AX','AV'],'W')
```

```
Tmax=15.0
NT = 3000
h = Tmax/NT
dirk = DIRK_Monolithic(h,LDIRK[1], sys,warp.update,apply_BCs,
                       warp.fields['wx'].vector(),warp.fields['wv'].vector(),
                       warp.assemble_form('M','W'))
# warp.CG.OutputFile("post/impact/gammaC.pvd" )
```

A.4 Do it:

And, finally, march forward in time:

```
for t in xrange(NT):
    if t%5==0:
        warp.create_contacts(pairs=cpairs,cutoff=1.5)
    dirk.march()
    if t%10==0:
        output()

embed()
```

Appendix B

Gaussian Quadrature on the Unit Disk

Since the fibrils analyzed were primarily circular in the cross section, integration across the unit disk was required. The integral over a section with radius R is of the form

$$\int\int_{R\geq\sqrt{X_1^2+X_2^2}} f\left(X_1, X_2\right) dX_1 dX_2.$$

To transform it to the unit disk, the uniform mapping

$$\begin{aligned} X_1 &= R\xi_1 \\ X_2 &= R\xi_2 \end{aligned}$$

is used, whose determinant is

$$J_\xi = \det\left(\frac{\partial \mathbf{X}}{\partial \boldsymbol{\xi}}\right) = R^2.$$

The integral can then be transformed as

$$\int\int_{1\geq\sqrt{\xi_1^2+\xi_2^2}} f\left(R\xi_1, R\xi_2\right) J_\xi d\xi_1 d\xi_2.$$

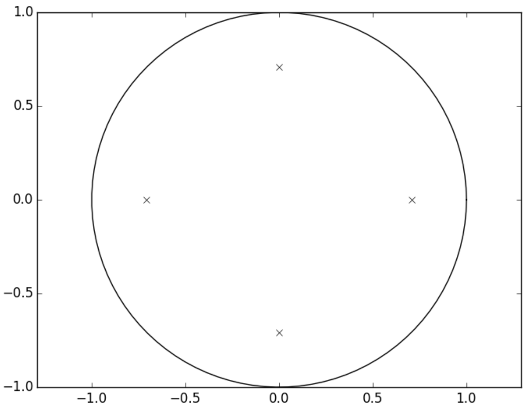
It can be calculated by performing a quadrature summation,

$$\sum_I w^I R^2 f\left(R\xi_1^I, R\xi_2^I\right)$$

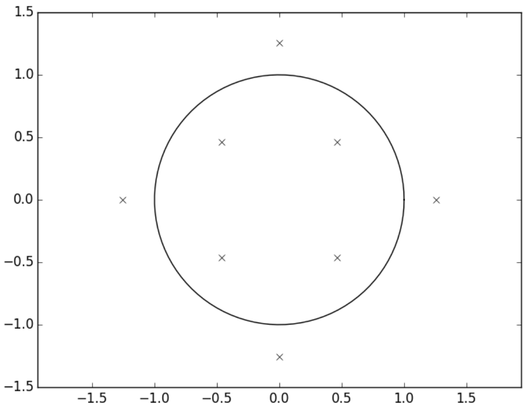
where w^I, ξ_1^I, and ξ_2^I are the weights and points. The rules used were obtained from [57] and are tabulated in Table B.1 alongside plots of their locations with respect to the unit disk. In the main library, the files “src/Forms/QuadraturePoints.py” and “src/Forms/QuadratureUnitTests.nb” perform tests on the quadrature rules to verify the accuracy of the rules compared to analytical integrals.

Table B.1: Table of quadrature points in polar coordinates and locations with respect to the unit disk.

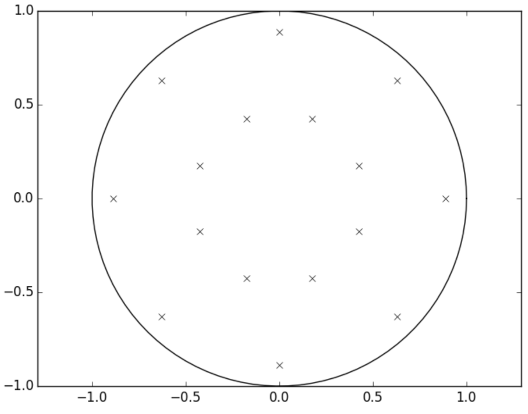
w^I	r^I	θ^I
$\frac{\pi}{4}$	$\frac{\sqrt{2}}{2}$	0
$\frac{\pi}{4}$	$\frac{\sqrt{2}}{2}$	$\frac{\pi}{2}$
$\frac{\pi}{4}$	$\frac{\sqrt{2}}{2}$	π
$\frac{\pi}{4}$	$\frac{\sqrt{2}}{2}$	$\frac{3\pi}{2}$



w^I	r^I	θ^I
$\pi\frac{2-\sqrt{3}}{16}$	$1+\frac{\sqrt{3}}{3}$	0
$\pi\frac{2-\sqrt{3}}{16}$	$1+\frac{\sqrt{3}}{3}$	$\frac{\pi}{2}$
$\pi\frac{2-\sqrt{3}}{16}$	$1+\frac{\sqrt{3}}{3}$	π
$\pi\frac{2-\sqrt{3}}{16}$	$1+\frac{\sqrt{3}}{3}$	$\frac{3\pi}{2}$
$\pi\frac{2+\sqrt{3}}{16}$	$1-\frac{\sqrt{3}}{3}$	$\frac{\pi}{4}$
$\pi\frac{2+\sqrt{3}}{16}$	$1-\frac{\sqrt{3}}{3}$	$\frac{3\pi}{4}$
$\pi\frac{2+\sqrt{3}}{16}$	$1-\frac{\sqrt{3}}{3}$	$\frac{5\pi}{4}$
$\pi\frac{2+\sqrt{3}}{16}$	$1-\frac{\sqrt{3}}{3}$	$\frac{7\pi}{4}$



w^I	r^I	θ^I	w^I	r^I	θ^I
$\frac{\pi}{16}$	$\frac{1}{2}+\frac{\sqrt{3}}{6}$	0	$\frac{\pi}{16}$	$\frac{1}{2}-\frac{\sqrt{3}}{6}$	$\frac{\pi}{8}$
$\frac{\pi}{16}$	$\frac{1}{2}+\frac{\sqrt{3}}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{16}$	$\frac{1}{2}-\frac{\sqrt{3}}{6}$	$\frac{\pi}{4}+\frac{\pi}{8}$
$\frac{\pi}{16}$	$\frac{1}{2}+\frac{\sqrt{3}}{6}$	$\frac{\pi}{2}$	$\frac{\pi}{16}$	$\frac{1}{2}-\frac{\sqrt{3}}{6}$	$\frac{\pi}{2}+\frac{\pi}{8}$
$\frac{\pi}{16}$	$\frac{1}{2}+\frac{\sqrt{3}}{6}$	$\frac{3\pi}{4}$	$\frac{\pi}{16}$	$\frac{1}{2}-\frac{\sqrt{3}}{6}$	$\frac{3\pi}{4}+\frac{\pi}{8}$
$\frac{\pi}{16}$	$\frac{1}{2}+\frac{\sqrt{3}}{6}$	π	$\frac{\pi}{16}$	$\frac{1}{2}-\frac{\sqrt{3}}{6}$	$\pi+\frac{\pi}{8}$
$\frac{\pi}{16}$	$\frac{1}{2}+\frac{\sqrt{3}}{6}$	$\frac{5\pi}{4}$	$\frac{\pi}{16}$	$\frac{1}{2}-\frac{\sqrt{3}}{6}$	$\frac{5\pi}{4}+\frac{\pi}{8}$
$\frac{\pi}{16}$	$\frac{1}{2}+\frac{\sqrt{3}}{6}$	$\frac{3\pi}{2}$	$\frac{\pi}{16}$	$\frac{1}{2}-\frac{\sqrt{3}}{6}$	$\frac{3\pi}{2}+\frac{\pi}{8}$
$\frac{\pi}{16}$	$\frac{1}{2}+\frac{\sqrt{3}}{6}$	$\frac{7\pi}{4}$	$\frac{\pi}{16}$	$\frac{1}{2}-\frac{\sqrt{3}}{6}$	$\frac{7\pi}{8}+\frac{\pi}{8}$



Appendix C

Butcher Tableaus

The Butcher Tableaus of the Runge-Kutta methods used in this work are listed in this section, written in the notation

$$\begin{array}{c|c} c_i & A_{ij} \\ \hline & b_j \end{array}$$

The derivations of these methods can be found in [1], from where they were obtained. These are singly-diagonal methods: all of diagonal entries of the A table are equal, and all entries above the diagonal are zero.

The one-stage method, with $\mathcal{O}\left(h\right)$ convergence, where h is the time step size, is commonly known as Backward Euler and has the tableau:

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

The two stage method of this family has $\mathcal{O}\left(h^2\right)$ convergence and is:

$$\begin{array}{c|cc} 1-\frac{\sqrt{2}}{2} & 1-\frac{\sqrt{2}}{2} & 0 \\ 1 & \frac{\sqrt{2}}{2} & 1-\frac{\sqrt{2}}{2} \\ \hline & \frac{\sqrt{2}}{2} & 1-\frac{\sqrt{2}}{2} \end{array}$$

The three stage method has $\mathcal{O}\left(h^3\right)$ convergence and has the form

$$\begin{array}{c|ccc} \alpha & \alpha & 0 & 0 \\ \tau & \tau-\alpha & \alpha & 0 \\ 1 & b_2 & b_2 & \alpha \\ \hline & b_1 & b_2 & \alpha \end{array}$$

where the diagonal satisfies $0 = \alpha^3 - 3\alpha^2 + \frac{3}{2}\alpha - \frac{1}{6}$ and $\frac{1}{2} < \alpha < \frac{1}{6}$. The numerical value of this root was calculated to be $\alpha = 0.43586652150845899942$ using high-precision arithmetic in Mathematica of twenty decimal digits (i.e., this value has sufficient accuracy to be copy-and-pasted into an implementation.) The other constants are determined from α by $\tau = \frac{1+\alpha}{2}$, and $b_1 = -\frac{1}{4}\left(6\alpha^2 - 16\alpha + 1\right)$, $b_2 = \frac{1}{4}\left(6\alpha^2 - 20\alpha + 5\right)$. There is no fourth order accurate method in this family that has only four stages, so only the first three are used.

Appendix D

Beam Visualization

The finite element meshes used in the beam discretization are one dimensional, but represent three dimensional objects with an appreciable thickness. One dimensional meshes with director information are useful, but become far too cluttered for visualization interwoven microstructures, e.g. Figure D.0.1. For visualization purposes, a three dimensional volumetric mesh of tetrahedra is created. In addition to creating a solid mesh, the fields defined on the finite element mesh need to be projected to the new 3D mesh. This is accomplished by manually editing the coefficients to a new function space. The construction of discrete points around the surface is illustrated in Figure D.0.2. The pseudocode for creating the mesh is listed in Algorithm D.1.

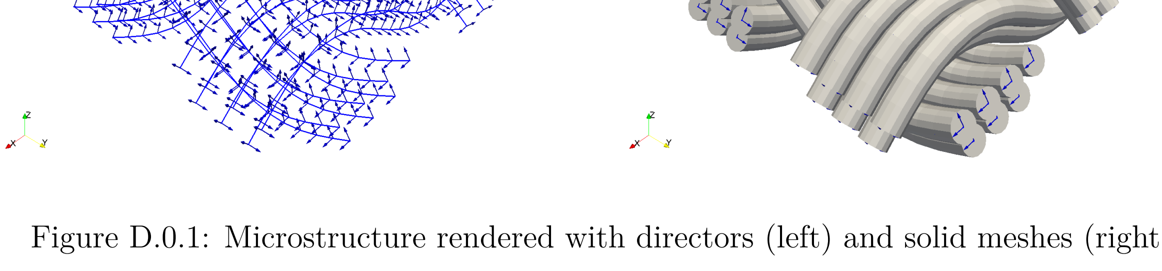
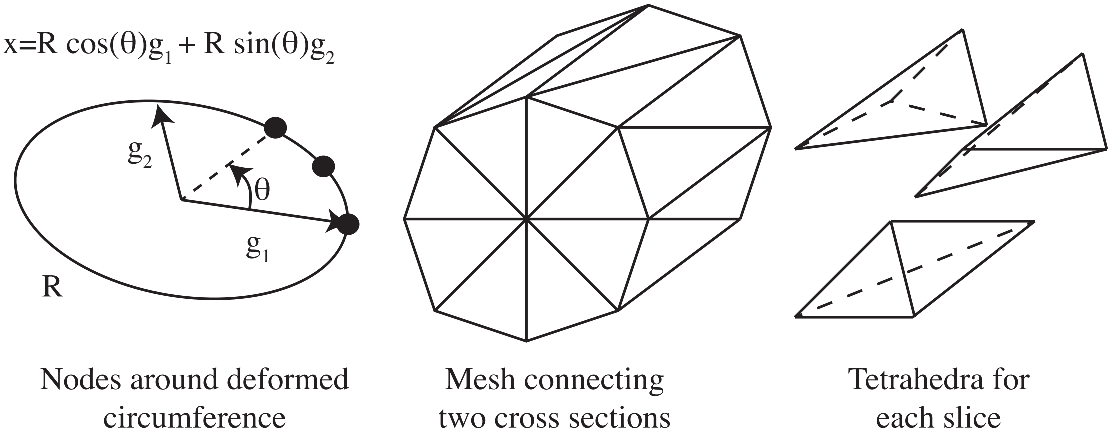


Figure D.0.1: Microstructure rendered with directors (left) and solid meshes (right)



Nodes around deformed circumference

Mesh connecting two cross sections

Tetrahedra for each slice

Figure D.0.2: Calculation of deformed surface geometry.

Algorithm D.1 Routine for generating 3D mesh. (Zero-based indices used)

```
Input the number of vertices to place around the surface, $n_\theta$, and a 1D mesh
  1. Project centroid deformation $\mathbf{q}$, and directors $\mathbf{g}_1$, and $\mathbf{g}_2$ to the vertices of the 1D mesh.
  2. Create an empty 3D mesh.
  3. Loop over the vertex indices $i \in 0, n_V$ of the 1D mesh:
     (a) Add $\mathbf{X0}[i] + \mathbf{q}[i]$ to the 3D mesh vertices
     (b) For $n_\theta$ equally spaced angles $\theta \in [0, 2\pi]$:
           i. Add $\mathbf{X0}[i] + \mathbf{q}[i] + R\cos(\theta)\,\mathbf{g}_1 + R\sin(\theta)\,\mathbf{g}_2$ to the 3D mesh vertices
  4. Loop over the vertices:
     (a) For $i \in 0, n_\theta$:
           i. Push three tetrahedra into the 3D mesh for the slice joining the two cross
              sections
  5. Finalize the 3D mesh
  6. Define a linear function space on the new 3D mesh
  7. Loop over all all fields in the output dictionary:
     (a) Create a new function on the mesh
     (b) Project the original function to the vertices of the 1D mesh
     (c) Manually set the degrees of freedom associated with all $n_\theta + 1$ vertices on the
         cross section
  8. Output the new fields to the file.
```